SANDIA REPORT

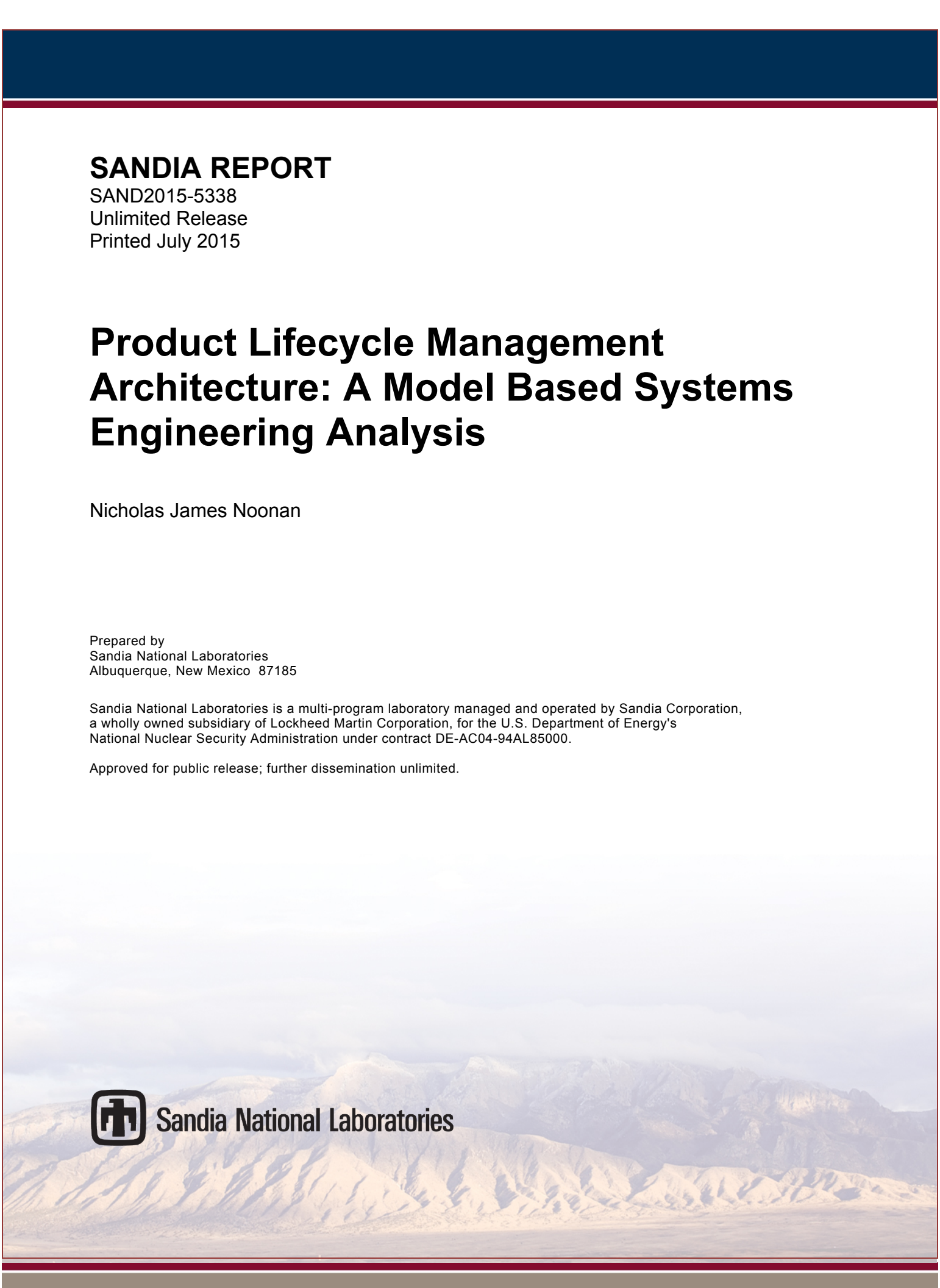
SAND2015-5338
Unlimited Release
Printed July 2015

Product Lifecycle Management Architecture: A Model Based Systems Engineering Analysis

Nicholas James Noonan

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: http://www.ntis.gov/search

SAND2015-5338
Unlimited Release
Printed July 2015

Product Lifecycle Management Architecture: A Model Based System Engineering Analysis

Nicholas James Noonan
Department 2982
Product Lifecycle Management
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS0624

Abstract

This report is an analysis of the Product Lifecycle Management (PLM) program. The analysis is centered on a need statement generated by a Nuclear Weapons (NW) customer. The need statement captured in this report creates an opportunity for the PLM to provide a robust service as a solution. Lifecycles for both the NW and PLM are analyzed using Model Based System Engineering (MBSE).

ACKNOWLEDGMENTS

[Include acknowledgments here, if any. Otherwise, leave this page blank.]

CONTENTS

FIGURES

NOMENCLATURE

BDD	Block Definition Diagram
CAB	Change Advisory Board
CI	Configuration Items
ConOps	Concept of Operations
DOD	Department of Defense
DOE	Department of Energy
IBD	Internal Block Diagram
ICAM	Integrated Computer-Aided Manufacturing
IDEF0	ICAM Definition
MBSE	Model Based System Engineering
NW	Nuclear Weapons Program
PDM	Product Definition Management
PLM	Product Lifecycle Management
RPP	Realize Product Procedure
SNL	Sandia National Laboratories
SysML	System Modeling Language

Page intentionally left blank.

1 INTRODUCTION TO PRODUCT LIFECYCLE MANAGEMENT ARCHITECTURE ANALYSIS

This report is an analysis of the Product Lifecycle Management (PLM) program. The analysis is centered on a need statement generated by a Nuclear Weapons (NW) customer. The need statement captured in this report creates an opportunity for the PLM to provide a robust service as a solution. Lifecycles for both the NW and PLM are analyzed using Model Based System Engineering (MBSE). The basis of this report is to introduce the need statement, methodology for a solution, details of the solution, and recommendations. The analysis highlights certain aspects of the NW architecture to demonstrate how the PLM architecture can enhance delivery of a product. The NW's 6.X architecture is discussed in parallel with the PLM architecture and identifies interfaces with existing Realize Product Procedure (RPP) requirements. In addition, new requirements are introduced as draft requirements for baseline PLM architecture. This report lays out the traditional 6.X architecture and highlights existing gaps in product definition/realization configuration management, requirements engineering, and system architecting. The architectural innovation suggested in this report starts at the NW 6.2 phase and works in parallel with the PLM architecture. The analysis highlights gaps in the current NW architecture, recommends a PLM architecture, provides details of a PLM architecture, develops draft baseline PLM requirements, and demonstrates Model Based System Engineering (MBSE).

1.1 Architectural Innovation

The basis for architectural innovation comes from several areas which include academic research and development, experience, and the possibility of implementing process improvement with minimal fiscal impact. Brief experience is defined as working within Engineering and Operations and being matrixed to the NW program for less than a year. Experience coupled with systems engineering academic opportunity in pursuit of a Master of Engineering in System Engineering through Stevens Institute of Technology has led to process improvement focus. To start this discussion, let's examine an article written in 1990 by Kim B. Clark from Harvard University and Rebecca M. Henderson from Massachusetts Institute of Technology titled, "*Architectural Innovation: The Reconfiguration of Existing Product Technologies and Failure of Established Firms."* Henderson and Clark begin by describing improving product designs and entirely new designs of a product and how each relates to organizational capabilities. Clark and Henderson

argue that, “Incremental innovation reinforces the capabilities of established organizations, while radical innovation forces them to ask a new set of questions, to draw on new technical and commercial skills, and to employ new problem-solving approaches.” Clark and Henderson’s report is focused on how innovations to a product change the way the product is linked but retains the core design concept, “and thus the basic knowledge underlying the components”. This component knowledge is referred to as architectural knowledge.

Clark and Henderson point out that the distinction between the product as a whole – the system – and the product in its parts – the components – has a long history in the design literature (but also in the NW product realization lifecycle). An interesting clarification illustrated by Clark and Henderson is:

> “The distinction between the product as a system and the product as a set of components underscores the idea that successful product development requires two types of knowledge. First, it requires component knowledge, or knowledge about each of the core design concepts and the way in which they are implemented in a particular component. Second, it requires architectural knowledge or knowledge about the ways in which the components are integrated and linked together into a coherent whole. The distinction between architectural component knowledge, or between the components themselves and the links between them, is a source of insight into the ways in which innovations differ from each other.”

The important point here is that Clark and Henderson are talking about two types of architectural changes: radical (new design) and incremental (change in component). Clark and Henderson argue that radical and innovative architectural changes are matters of degree and they both present a unique set of challenges. Challenges faced by radical innovation are described as, “… radial innovation creates unmistakable challenges for established firms, since it destroys the usefulness of their existing capabilities. In our terms, it destroys the usefulness of both architectural and component knowledge” (Clark and Henderson). Whereas challenges faced by incremental innovation are defined as, “… presents established firms with a more subtle challenge. Much of what the firm knows is useful and needs to be applied in the new product, but some of what is known is not only not useful but may actually handicap the firm. Recognizing what is useful and what is not, and acquiring and applying new knowledge when necessary, may

be quite difficult for an established firm because of the way knowledge – particularly architectural knowledge – is organized and managed" (Clark and Henderson).

> "An organization's communication channels, both those that are implicit in its formal organization (A report to B) and those that are informal ("I always call Fred because he knows about X"), develop around those interactions within the organization that are critical to its task (Galbrath, 1973 Arrow, 1974). These are also the interactions that are critical to effective design. They are the relationships around which the organization builds architectural knowledge. Thus and organization's communication channels will come to embody its architectural knowledge of the linkages between components that are critical to effective design."

To further emphasize this point, an established organization will amplify its architectural knowledge by solving various types of problems until a comprehensive level of experience exists and no "new" problems present themselves. As Clark and Henderson point out, "In effect, an organization's problem-solving strategies summarize what it has learned about fruitful ways to solve problems in its immediate environment" (March and Simon, 1958; Lyles and Mitroff, 1980, nelson and Winter, 1982). For the purpose of this study, the design and engineering work currently scoped out to the NW is categorized as incremental architectural innovation.

1.2 Sandia National Laboratories and Incremental Architectural Innovation

As new technology emerges and as NW weapon systems are analyzed, new functionality has the potential to become part of the NW program. New technology and component functionality will either be an extension of a current mission or a redesign for an alternate mission. As it stands, when an existing nuclear weapon has upgrades to its components, enhancements to its capabilities, or changes in its mission, the core design remains the same. The core design in the context of this report of a nuclear weapon is the big bang (intended yield), surety, and safety. The architectural knowledge in relation to NW programs is the well-established 6.X process. The current 6.X process is defined as the Product Realization Lifecycle and includes seven phases which are represented in Figure 1.

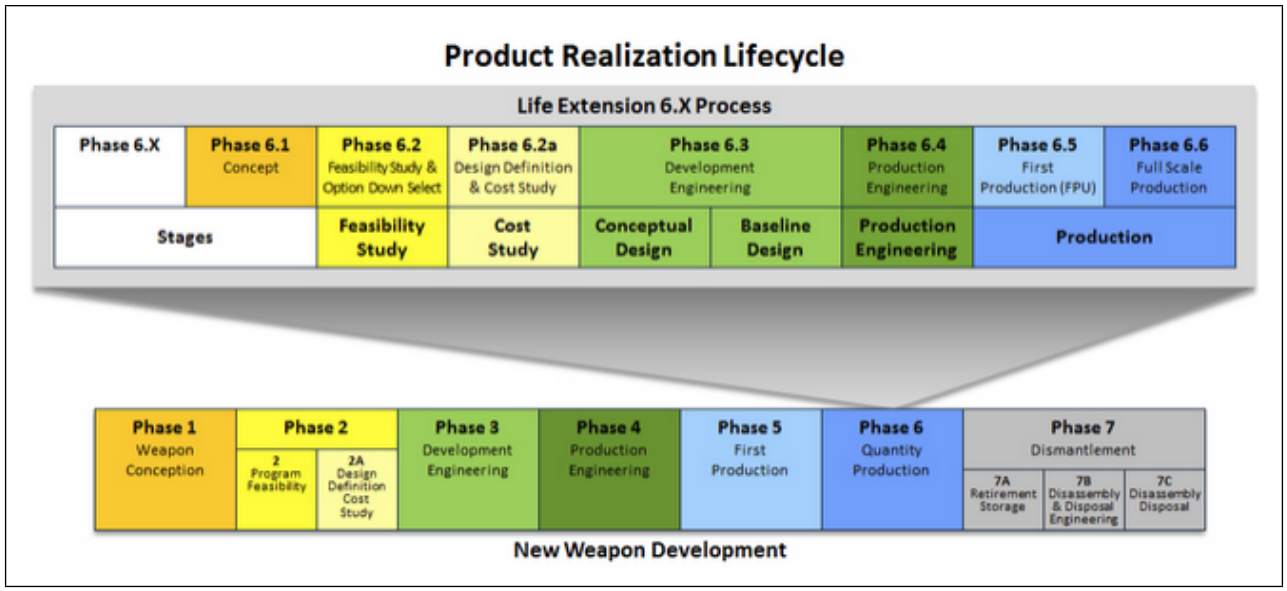


Figure 1. Nuclear Weapon Product Realization Lifecycle

As Figure 1 shows, a very well defined architectural knowledge has been established within Sandia National Laboratories and shared among external nuclear complex partners. A series of RPPs have been published (150 documents) outlining various processes relating to each phase of the 6.X process. The challenges faced by each nuclear weapon to undergo the 6.X process is that component technology changes (sometimes parts get smaller and provide more functionality) along with changes to process technology (e.g. platforms for computer-aided design are integrated to product structure). Industry capabilities are introduced to SNL and an attempt is made to integrate process technology with the intent of boosting design efficiencies and effectiveness. However, with the introduction to new component technology and process technology, incremental architectural innovations are not recognized. Both component technology and process technology are considered innovative architectural changes. As Clark and Henderson explain, "But information that might warn the organization that a particular innovation is architectural may be screened out by the information filters and communication channels that embody old architectural knowledge". This particular sentence is applicable to the 6.X process because incremental architectural innovation exists within the 6.X process and within SNL. Information that should be considered as warning to SNL that incremental architectural innovation is needed can be found in the very technology used within the NW program and in the processes which are followed to complete each program lifecycle. This report proposes that the existing 6.X Product Realization Lifecycle architecture should interface and

work in parallel with a Product Lifecycle Management architecture. The main purpose of this concept is keeping the 6.X Product Realization Lifecycle architectural knowledge aware of incremental architectural innovations and direct changes through a Change Advisory Board (CAB).

1.3 Mapping of Nuclear Weapon Product Realization Lifecycle to Product Lifecycle Management Architecture

As the 6.X process continues to execute large NW programs, component knowledge is changing, and in parallel, process technology is changing. These changes have resulted in trouble areas and have cost each program unknown amounts of time, money, and resources just in data management. Specifically, these trouble areas can be grouped into product structure, configuration management, systems architecting, and requirements engineering & management. These trouble areas are not areas that have failed in themselves alone, but are affects initiation by component knowledge design changes and should have triggered incremental architectural innovations. As a result of challenges facing the nuclear weapons programs, a needs statement was developed by a team within an existing nuclear weapon program. For the purpose of this report, the needs statement has been slightly updated to fit the context of the program it originated from. The needs statement is captured in Section 3 of this report. This report intends to not only to bring attention to architectural innovation, but to also provide an appropriate analysis of the 6.X process and PLM process including a comprehensive and detailed solution. The solution space is the evidence for supporting a future state of PLM in which specific services can be deployed to a customer. Recommended support offered by the PLM includes planning early-on in the areas of configuration management, product definition architecture, system architecture, and requirements engineering & management.

2 METHODOLOGY

As mentioned in Section 1, MBSE is used to convey the concepts of this report. The software used to develop this analysis was a Vitech product (CORE 9.0 Spectrum) and a software program called Astah.

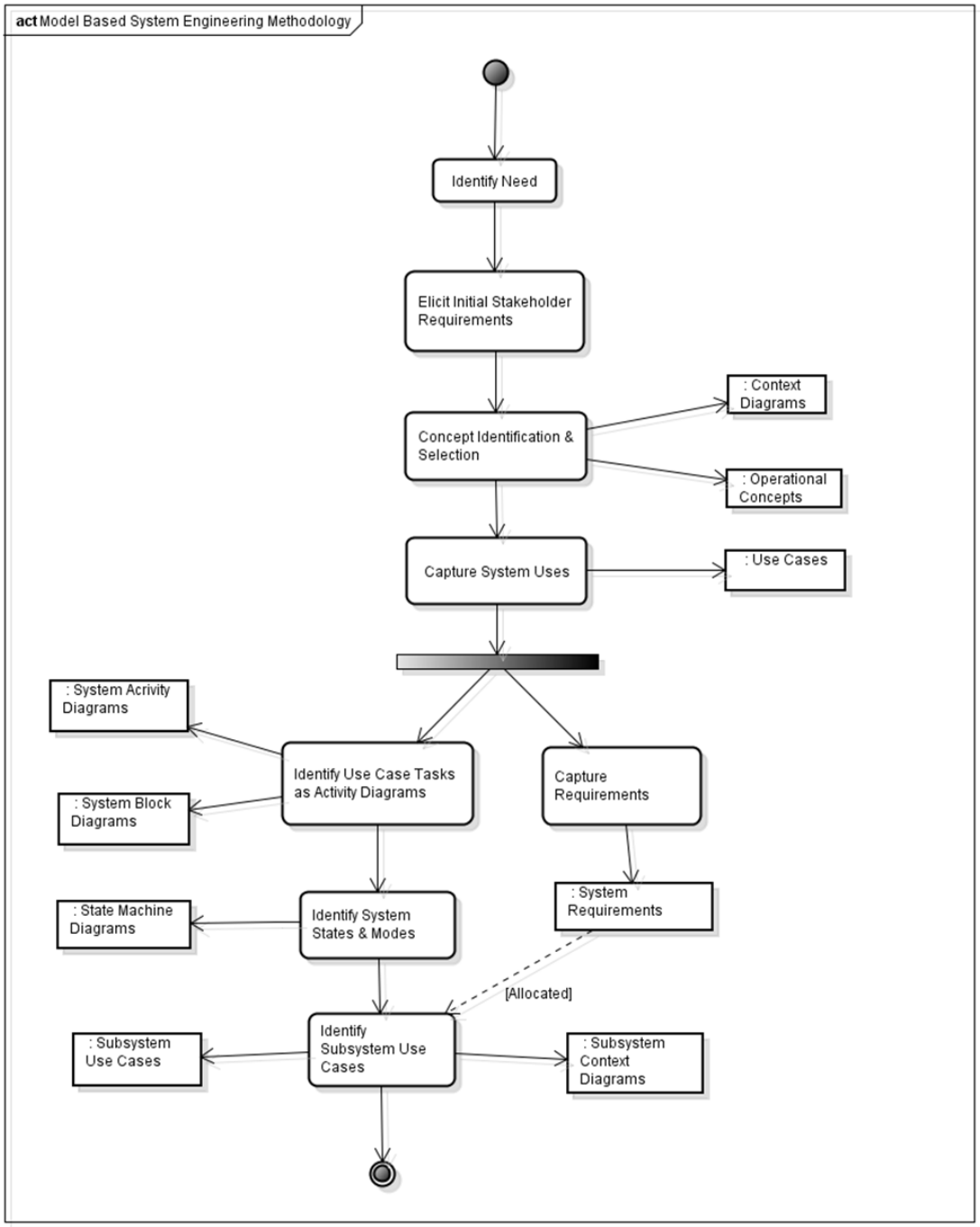


Figure 2. Methodology Used in This Report

As shown in Figure 2, the methodology followed in this report begins with the identification of a need, followed by the elicitation of stakeholder requirements, creation of a concept, and analysis of use cases. Once the use cases have been created, further requirements are developed which are allocated to the system, which in this case is the PLM architecture. As Buede describes, "The allocated architecture integrates the requirements decomposition with the functional and physical architecture" (Buede, 2009). A distinct difference of the allocation development mentioned by Buede is that in this analysis of the PLM, the use of a logical architecture takes place of the mentioned functional architecture. Further details and descriptions of the allocated architecture development are found in Section 7. Each SysML diagram used in this report is defined below. According to authors Dennis Buede and Lenny Delligatti, the definitions are summarized as:

- **Block Definition Diagram (BDD**) - "Used to display blocks such as blocks and value types and the relationships between those blocks" (Delligatti, 2014).
- **Internal Block Diagram (IBD)** - "Used to specify the internal structure of a single block. More precisely, an IBD shows the connections between the internal parts of a block and the interfaces between them" (Delligatti, 2014).
- **Use Case Diagram** - "Used to convey the use cases that a system performs and the actors that invoke and participate in them" (Delligatti, 2014).
- **Activity Diagrams** - "Used to specify a behavior, with a focus on the flow of control and the transformation of inputs into outputs through a sequence of actions" (Delligatti, 2014).
- **Sequence Diagrams** - "Used to specify a behavior, with a focus on how the parts of a block interact with one another via operational calls and asynchronous signals" (Delligatti, 2014).
- **Parametric Diagrams** - "Used to express how one or more constraints - specifically, equations and inequalities - are bound to the properties of a system" (Delligatti, 2014).

Non-SysML Diagrams include:

- **IDEF0** - "IDEF0 acronym comes from the U.S. Air Force's Integrated Computer-Aided Manufacturing (ICAM) program that began in the 1970's. IDEF is a complex acronym that stands for ICAM Definition. The number, 0, is appended because this modeling

technique was the first of many techniques developed as part of this program" (Buede, 2009).

- **Hierarchy Diagram** - Used by CORE to view a variety of pre-defined or custom diagrams. This diagram is helpful when demonstrating traceability of architecture allocation or requirements.

3 IDENTIFY NEED

The need statement described in this section originates from a real-world situation where a program was faced with various execution problems relating to configuration management, tool usage, collaboration rules and best practices, and archiving data. The process to define a need statement, recommendations, and actual execution of a resolution is time-consuming for a program to undertake during a design engineering phase. The purpose of this report is to use the developed need statement as evidence that PLM services can be provided to programs early in the initiation phase, which can alleviate root causes of many of the identified issues.

3.1 Situation Background

The need statement identified in this report was the result of several years of uncoordinated attempts to implement configuration management and design collaboration. The challenges to correct these uncoordinated issues of change control and development work resulted in the dismissal of a Product Definition Management (PDM) tool and an implementation tasking to the way things were previously done. Previous programs have struggled with extremely similar issues but were never canceled or resulted in design failures. An attempt to implement new PDM capabilities was done during the development engineering phase (6.3), which is too late in a program life-cycle to have the desired effectiveness. Therefore, when personnel on a program are asked to change their work habits to a foreign practice without enforcement or accountability, resistance is ubiquitous - especially since past programs have managed in the past.

3.2 Need Statement

With the exception of modeling and design documents generated by the design group, there is a lack of accepted processes, practices or storage locations for managing documents associated with the development engineering phase of the Nuclear Weapon 6.X Process. As a result, team

members store documents in multiple locations with little consistency, struggle to locate what they need quickly and easily and often don't know whether or not they are referencing the record copy. In addition, the program faces a potential vulnerability should there be an audit of the nuclear weapon design documentation and procedures related to them. A need exists for guidance to be established of how program personnel store documents, what tools to use when collaborating, how to submit program changes, track changes, review and release finalized work, and how to search and access development work in progress.

3.3 Product Lifecycle Management and Need Statement

Due to the creation of the need statement above, a gap is identified by how programs have historically been executed when following the 6.X process. The gap is the role of the PLM department. Over time, this gap has grown due to changes in technology primarily with solid modeling and managing the exponential increase of data generated when documenting design decisions. As technology changes, solid modeling is becoming ever more incorporated with product structure, configuration management, verification and validation, internal collaborating, and external collaborating. Because of the change in how programs execute development engineering, PLM needs to become a larger stakeholder in programs lacking the qualified and needed resources. A PLM department at SNL should be able to deploy services to any program required to process through the 6.X Product Lifecycle Realization. Services that the PLM should deploy are:

- Configuration Management
- Product Definition & Realization
- Requirements Engineering and Management
- System Architecture

3.3.1 Configuration Management

Configuration management services provided by the PLM can be created specifically for a customer need, but some PLM requirements remain standard.

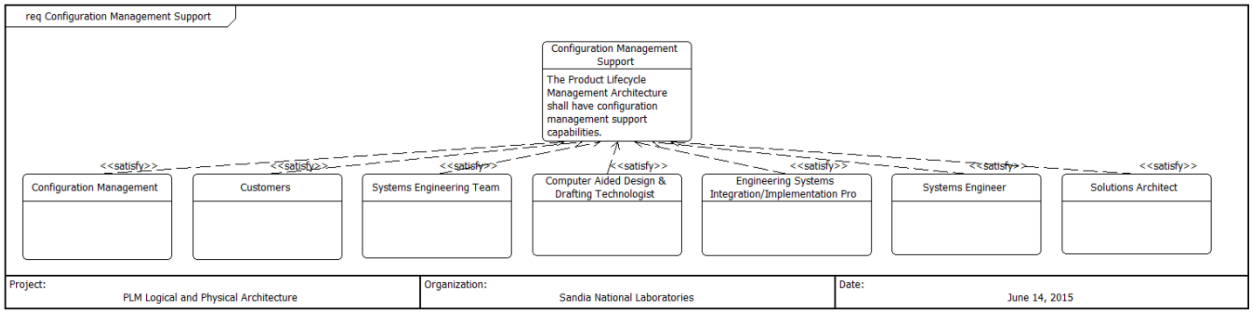


Figure 3. Baseline Example of Configuration Management Requirement

For an example, Figure 3 shows how an PLM high-level configuration management requirement applies to multiple PLM logical architecture blocks before any customization is applied.

Requirement Statement:

The Product Lifecycle Management Architecture shall have configuration management support capabilities.

Refines Higher-Level Requirement:

- NWP.1 Product Lifecycle Management Architecture Authorization

  -Basis Of:

  - PLMLog.1.2 Configuration Management
  - PLMLog.1.3 Customers
  - PLMLog.2.16 Systems Engineering Team
  - PLMLog.3.1 Computer Aided Design & Drafting Technologist
  - PLMLog.3.4 Engineering Systems Integration/Implementation Pro
  - PLMLog.3.7 Systems Engineer
  - PLMLog.3.8 Solutions Architect

Each requirement refines a higher-level requirement and is allocated to the logical and physical architecture. In the example above, the configuraiton management requirement is the basis of the logical architecture and refines the higher-level NWP.1 requirement. Configuration management is an absolute necessity for any PLM architecture and it must be adopted early on in any program lifecycle. Further allocation architecture development is detailed in Section 7.

3.3.2 Product Definition & Realization

Product definition & realization services provided by the PLM can also be customized based on the need of the customer. In general, the PLM should have a well-defined process for supporting a customer's product realization needs throughout the product lifecycle. What currently lacks

with product definition and realization is a full understanding of how important this service is and how interrelated with solid modeling and other blocks it becomes during the development process. By identifying the importance of product definition early in the 6.X process, the PLM will be able to support all of a program's needs to include quality management and retirement.

3.3.3 Requirements Engineering and Management

One of the main gaps identified in this analysis is the lack of a deployable team of requirements specialists. A PLM with a qualified and deployable team of requirements engineering and management helps with configuration management, product definition & realization, and system architecting. Requirements are often tied to program milestones, design decisions, and product realization strategies.

3.3.4 System Architecture

Like requirements engineering and management, system architecture is often undervalued and not fully considered when programs are first undergoing initiation and planning phases. System architecture provides the foundation for integration, verification, and validation. In addition, system architecture helps determine early on the engineering of a system. Buede describes this process as, "**Engineering of a System:** engineering discipline that develops, matches, and trades off requirements, functions, and alternate system resources to achieve a cost-effective, life-cycle-balanced product based upon the needs of the stakeholders" (Buede, 2009). What Buede is describing is in relation to the typical Vee model, where a horizontal line is drawn under the intersection of the Vee. This line, "depicts the hand off of the final products of the design process, the CI specs, to the discipline (or design) engineers, those engineers whose orientation is electrical, mechanical, chemical, civil, aerospace, computer science, and the like whose job it is to produce a physical entity" (Buede, 2009). This is what system architecture offers from a PLM perspective along with a system design which is used throughout the program lifecycle for communication and clarification.

4 SOLUTION SPACE: INITIAL STAKEHOLDER REQUIREMENTS

Initial stakeholder requirements for the PLM are based on the stakeholder orientation. For the purpose of this report, stakeholders are categorized as either active or passive.

- Active Stakeholders: Individuals, entities, other systems which will actively interact with the "system" once it is operational and in use. (source)
- Passive Stakeholders: Individuals, entities, other systems, standards, protocols, procedures, regulations, that will also influence the success of the system or solution.
    - Example: regulatory bodies; government agencies (source)

To begin with, the nuclear weapon program is subjected to various stakeholder requirements (both active and passive). The PLM is also subjected to various stakeholder requirements (both active and passive). However, the stakeholder interaction varies from NW programs to PLM programs. NW programs are directly subjected to a large array of passive stakeholder requirements which often makeup a complex set of origination requirements. Figure 4 shows various levels of stakeholders for an NW program. Level 0 – 3 are originating passive stakeholder requirements. Level 4 is where SNL has created Realize Product Procedure (RPP) requirements to help the NW program ensure they satisfy the higher-level stakeholder requirements. When engaged with NW programs, it is essential that the PLM clearly define stakeholders and their requirements. Since the PLM has a different domain than NW programs, some of the requirements will not apply while others are directly imposed upon the PLM.

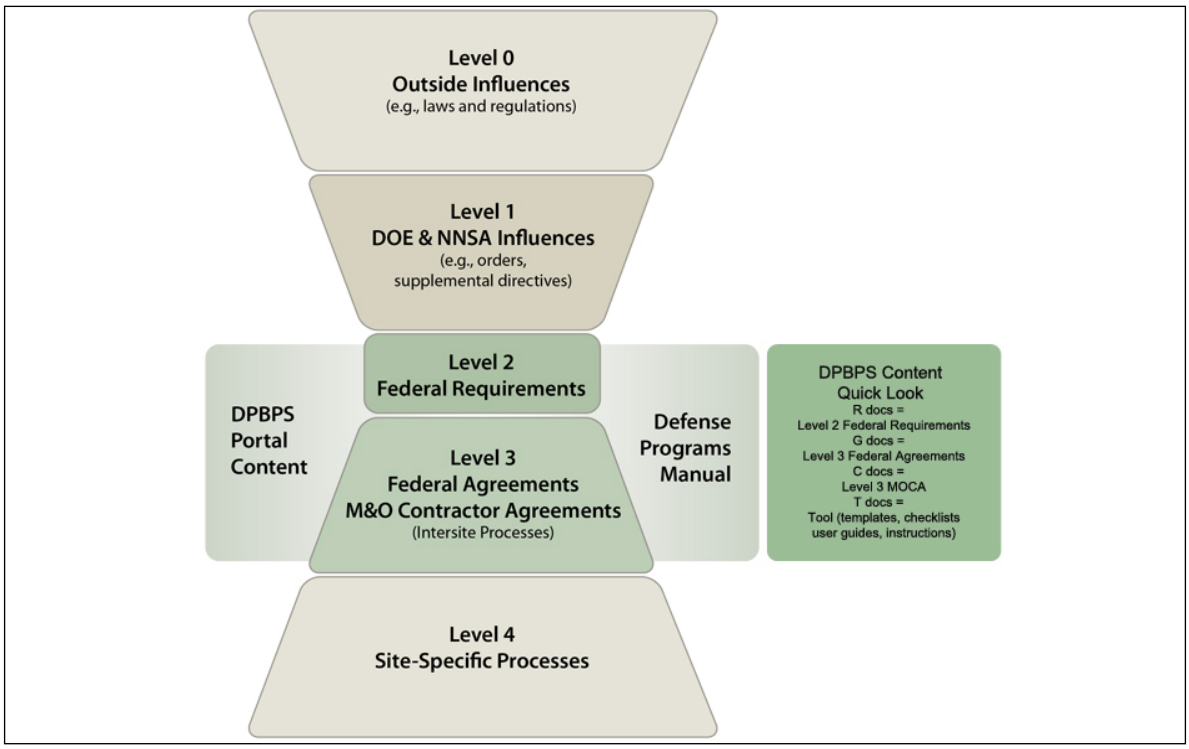


Figure 4. Nuclear Weapon Stakeholder Derivation Diagram

In order to understand which NW programs requirements are imposed on the PLM, a stakeholder diagram must be developed and used as a baseline. Figure 5 is a stakeholder diagram showing active and passive stakeholders for the PLM. By analyzing NW program requirements, the PLM has an opportunity to create a baseline set of originating stakeholder requirements.

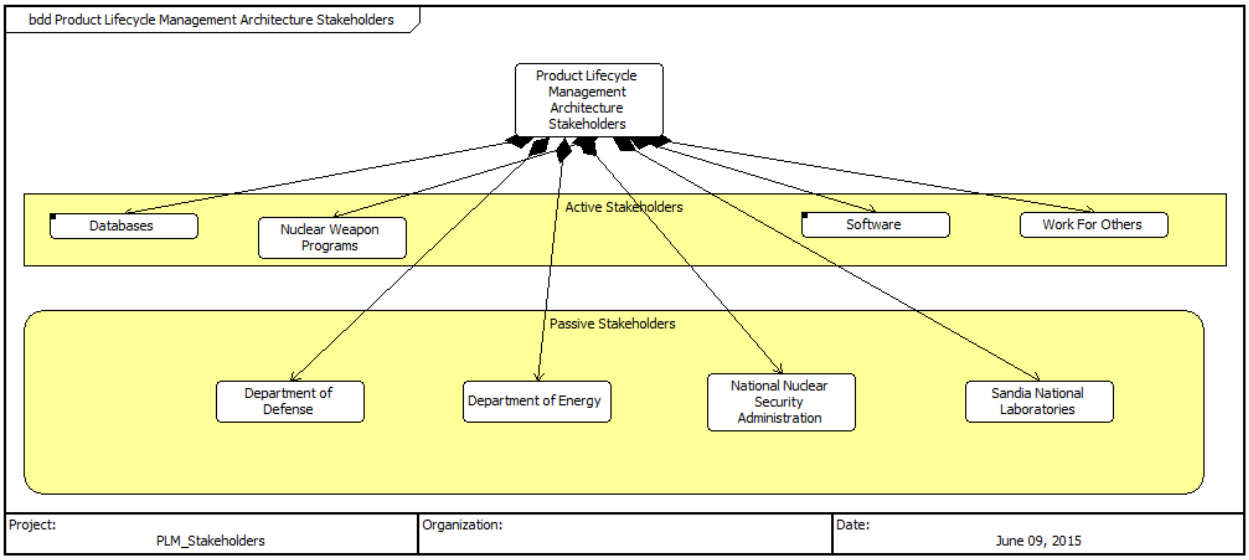


Figure 5. Block Definition of the Product Lifecycle Management Stakeholders

A passive stakeholder requirement for the PLM is one in which the requirement will shape the design of the architecture and determine at what point PLM interaction with the customer will take place. A good example of this is the federal requirement for the NW program – **R0001 Product Realization**. The scope of this document is explains:

"This content is used for WR and JTA product realization. The IPG requirements herein allow tailoring as determined by the project Team and are included in an IPG-IP per T140. For 6.X refurbishment programs, additional requirements are defined in R006."

This document contains over 80 requirements, some of which will not apply to the PLM, some of which will directly influence the 6.2 phase. Other examples of federal requirement documents are:

- **R003 – Product Definition Control**
- **R006 – 6.X Process**
- **R008 – Portfolio-Program-Project Management**
- **R012 – Requirements Engineering**

Related to Figure 4, level four, the PLM should be subjected to a large amount of active stakeholder originating requirements.

- Examples of relevant RPPs:
    - **RPP - 5 Configuration Management Plan for PRTs**
    - **RPP - 113 Software Product Engineering and Qualification**
    - **RPP - 301 Military Characteristics (MC) and Stockpile-to-Target Sequence (STS) Documents**
    - **RPP - 302 Mechanical Computer-Aided Design (MCAD)**
    - **RPP - 305 Interface Control Documents**
    - **RPP - 308 Requirements Management**
    - **RPP - 302 Procurement Index**
- In addition, there are multiple product realization standards which are also applicable to the PLM architecture. Below are a few examples:
    - **PRS11101 - Electronic File Transfer of the Pro/Engineering Design Definition Set**
    - **PRS12003 - Configuration Management for Mechanical CAD Files**

As Figure 5 shows, software is considered an active stakeholder due to the level of requirements and the need to customize each architecture to best represent a customer's need. Figure 6 shows an expansion of software stakeholder for an NW program. The PLM architecture must include specific software capabilities in order to satisfy origination of stakeholder requirements. In addition to software tools, the PLM must employ qualified resources who can be deployed to each program and be able to utilize software displayed in Figure 6.

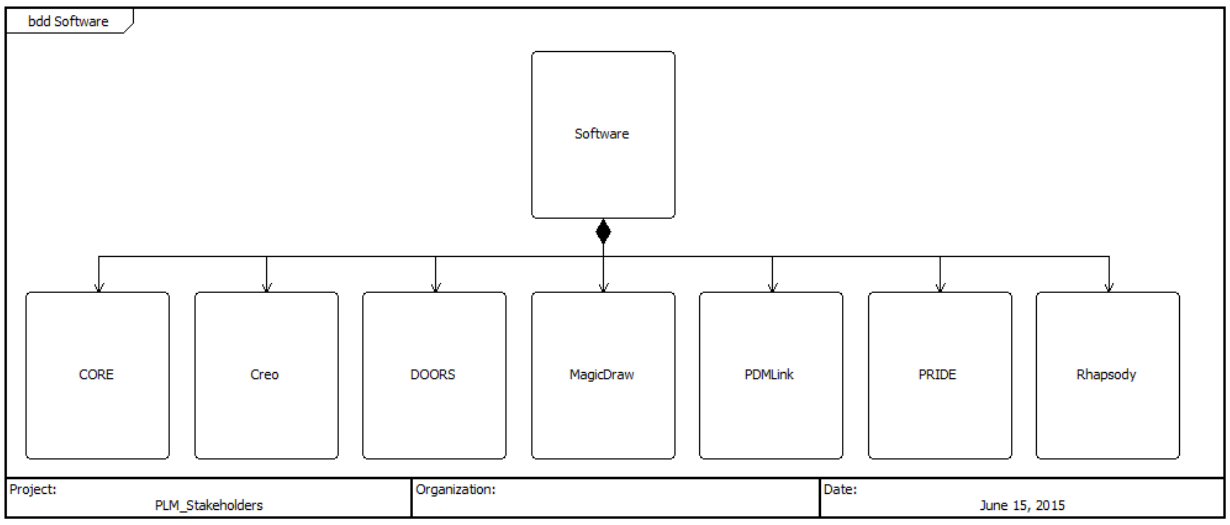


Figure 6. Expansion of Software Stakeholders for Product Lifecycle Management Architecture

5 SOLUTION SPACE: SOLUTION CONCEPT

The concept explained in this report is based on a Product Lifecycle Management methodology which is to be the governance body for Product Lifecycle Management at SNL. The Product Lifecycle Management mission for the domain is:

> *"The governing body that standardizes Sandia's engineering resources to enable lab-wide efficiencies, a disciplined approach to engineering, and an integrated engineering environment with easy access to tools and capabilities."*

By modeling current and future resources along with processes, the reality of the PLM becoming a governing body for SNL can be communicated. Most importantly, modeling allows for an in-depth analysis into various processes resulting in a complete and traceable set of requirements.

A service based PLM concept offers a solution not only to the need statement discussed in Section 3.2, but it allows customers the options to employ professionals uniquely situated to provide value when needed. In addition, having a PLM independent from a large program such as nuclear weapons gives the PLM management the opportunity to explore process improvement. PLM process improvement comes from leveraging capabilities from within SNL and industry alike. This section discusses the purpose and benefits of modeling and offers several examples of analysis capabilities. Section 5.2 begins the discussion of the PLM concept of operations (ConOPs) in relation to nuclear weapons and the 6.X process. The ConOps becomes a vital part of the model because it describes the starting and ending relationships between the PLM and a customer. Furthermore, the ConOps helps to discuss the level of detail the PLM

should engage with a customer's business process and can be used to cover the solution for the needs statement discussed in Section 3.2.

5.1 Purpose of the Product Lifecycle Management Model

The purpose of modeling the PLM includes:

1. Analyzing current capabilities
2. Analyzing current customer interactions and interfaces
   2.1. Analyzing process flows
   2.2. Identifying process gaps
   2.3. Identifying roles and responsibilities
3. Understanding where the PLM can offer services and when to "hand-off" services to other departments
4. Creating a baseline set of requirements to build and improve upon
5. Creating a PLM architecture to invoke argumentation which otherwise would have never been considered
6. Enhancing current and future capabilities by implementing widely accepted best practices

5.1.1 Model Based System Engineering - PLM Simulation Analysis Example

One of the benefits of MBSE is offering views of processes and, in some cases, running simulations to explain inefficiencies.

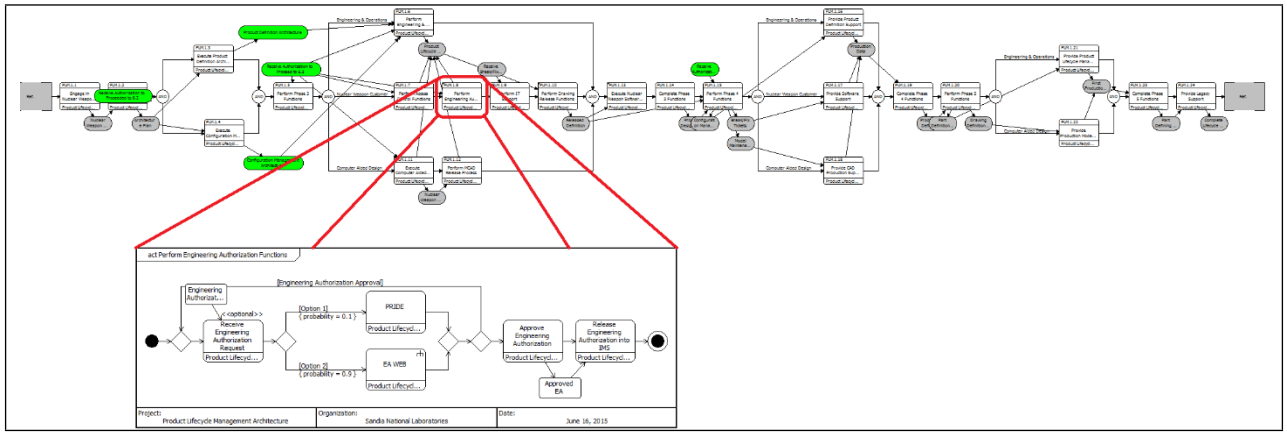


Figure 7. Activity Diagram and Decomposed Functions

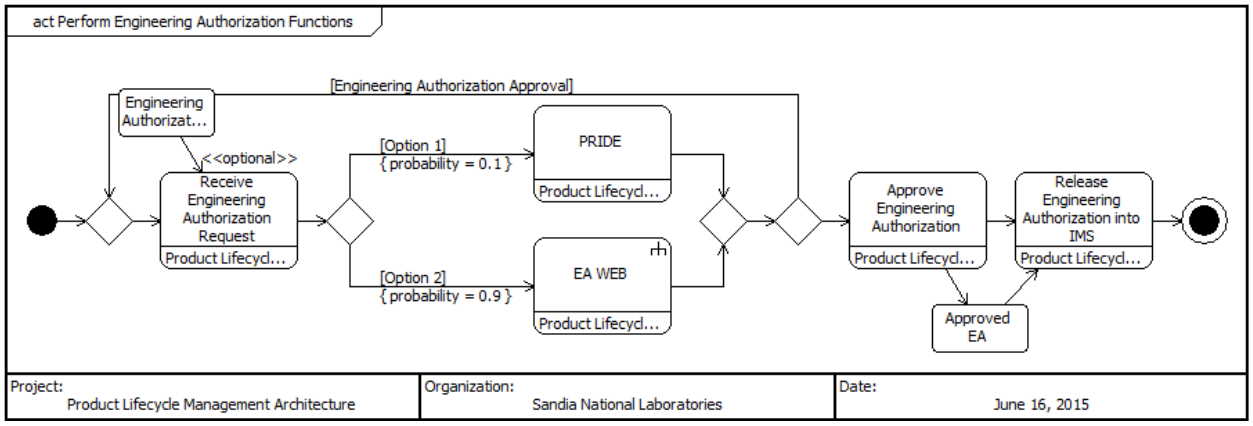


Figure 8. Example of Activity Diagram and EA WEB at 90% Utilization

Figure 8 shows an example of an activity diagram showing 10% probability of work going through PRIDE and 90% probability of work continuing to process through EA Web.

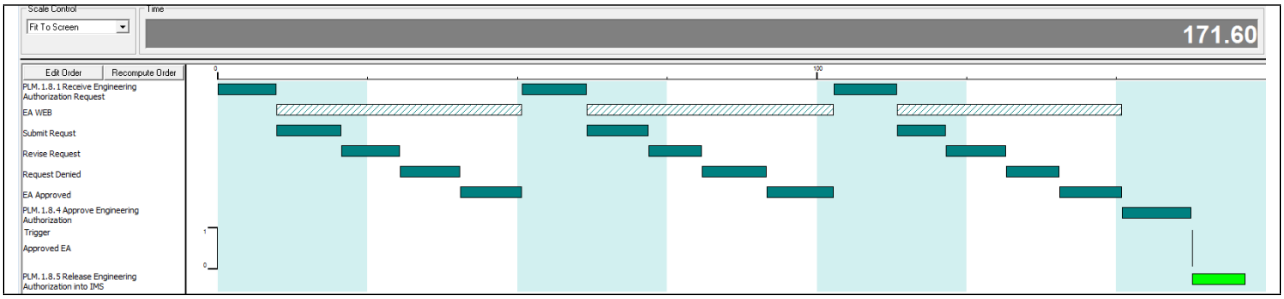


Figure 9. Example of Simulation for EA WEB Process

Figure 9 shows a probability that 90% of work will process through the EA WEB process. The result is that the EA WEB process will operate at 171.60 units.

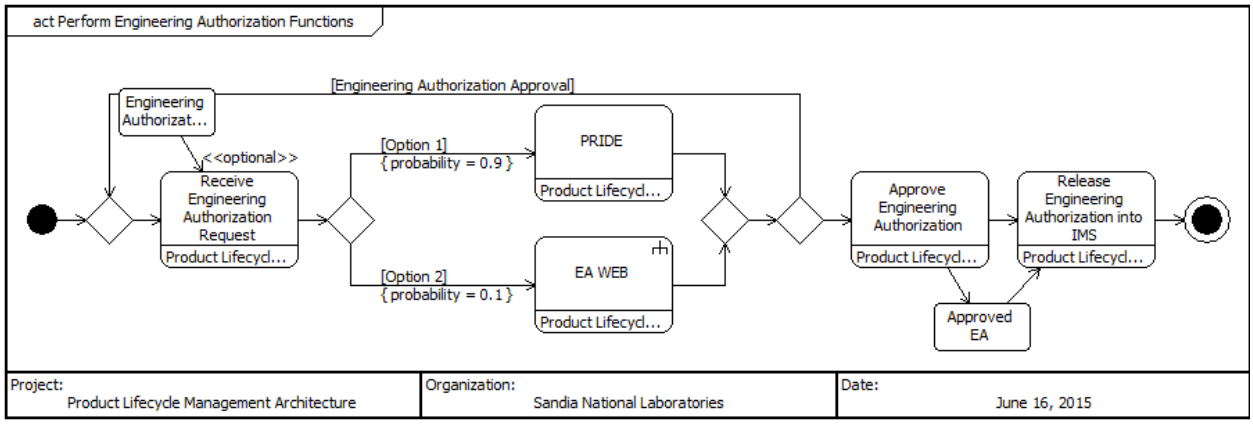


Figure 10. Example of Activity Diagram and PRIDE at 90% Utilization

Figure 10 shows a swap from the probabilities where 90% of work processes through PRIDE and 10% process through EA WEB.

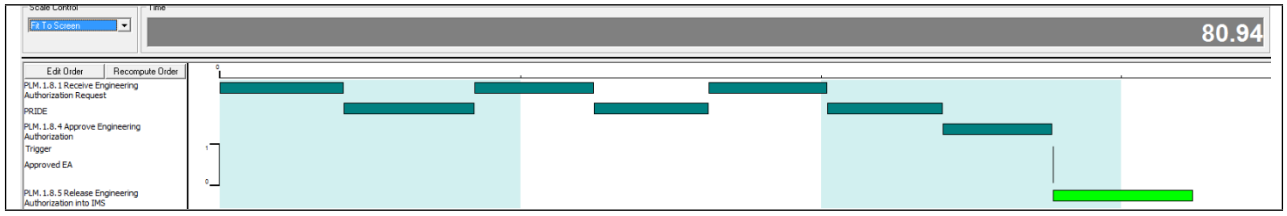


Figure 11. Example of Simulation for PRIDE Process

Figure 11 shows a simulation demonstrating that PRIDE processing is much more efficient at 80.9.

5.1.2 Model Based Systems Engineering - Process Creation Example

Figure 12 and Figure 13 are examples of how to analyze and create a new process using IDEF0. As Figure 12 shows, level 1 of the process is the high level showing inputs and outputs. Figure 13 is a decomposition showing the process functions. Having the ability to decompose functions in a process allows for the continued development of threads or activity diagrams. From this point forward, the process can be documented resulting in a report which serves as the basis for creating a new process. The IDEF0 chart shows the items, mechanisms, controls, and outputs from the process. All steps of the process can be verified (or further developed) in a detailed report.

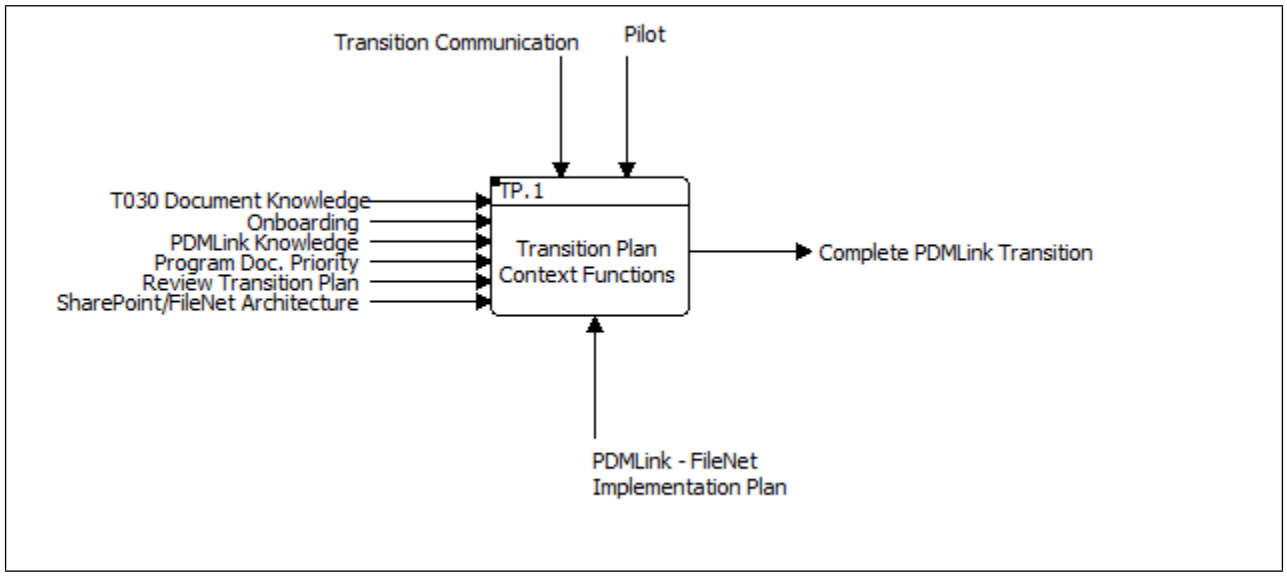


Figure 12. Example of IDEF0 Level 1 Process

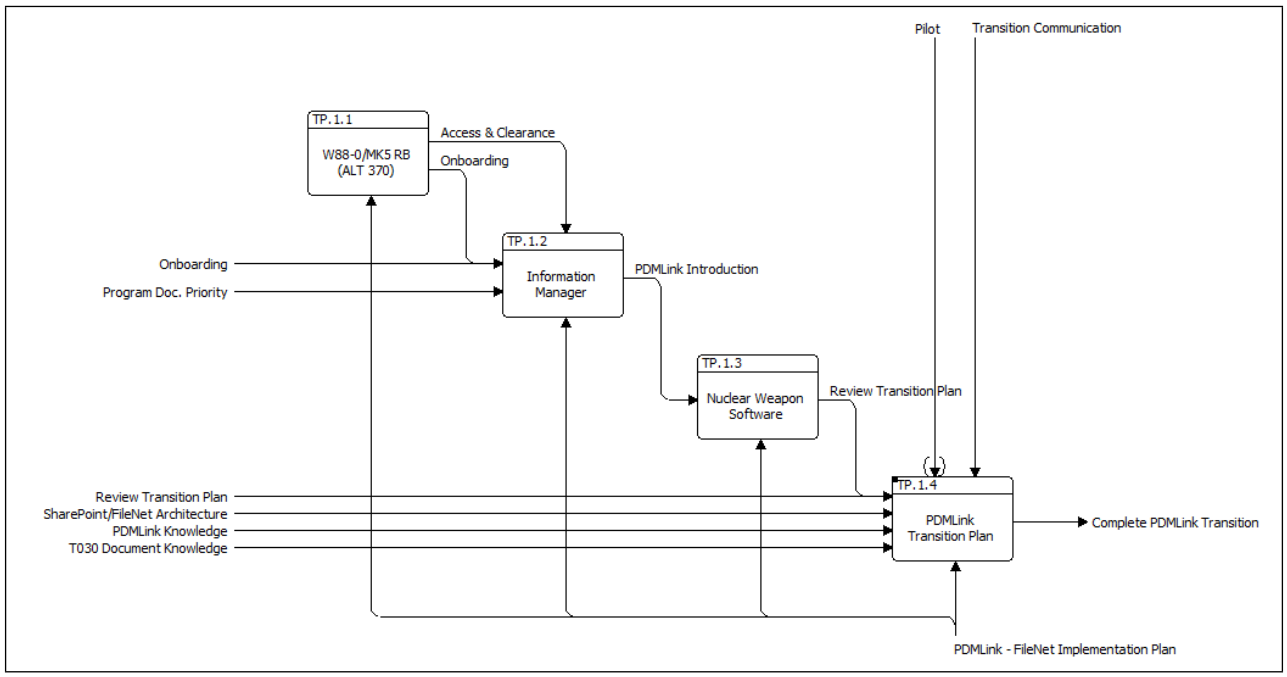


Figure 13. Example of IDEF0 Level 2 Process

5.1.3 Model Based Systems Engineering - Parametric Example

Another advantage of using MBSE is having the capabilities to produce parametric diagrams. As Delligatti explains, "The parametric diagram is a unique kind of SysML diagram, one that's used to express information about a systems constraints" (Delligatti, 2014). Figure 14 is an example of a block definition to show where constrains originate from for personnel and support bandwidth for the PLM. Figure 15 is an example of the parametric diagram.

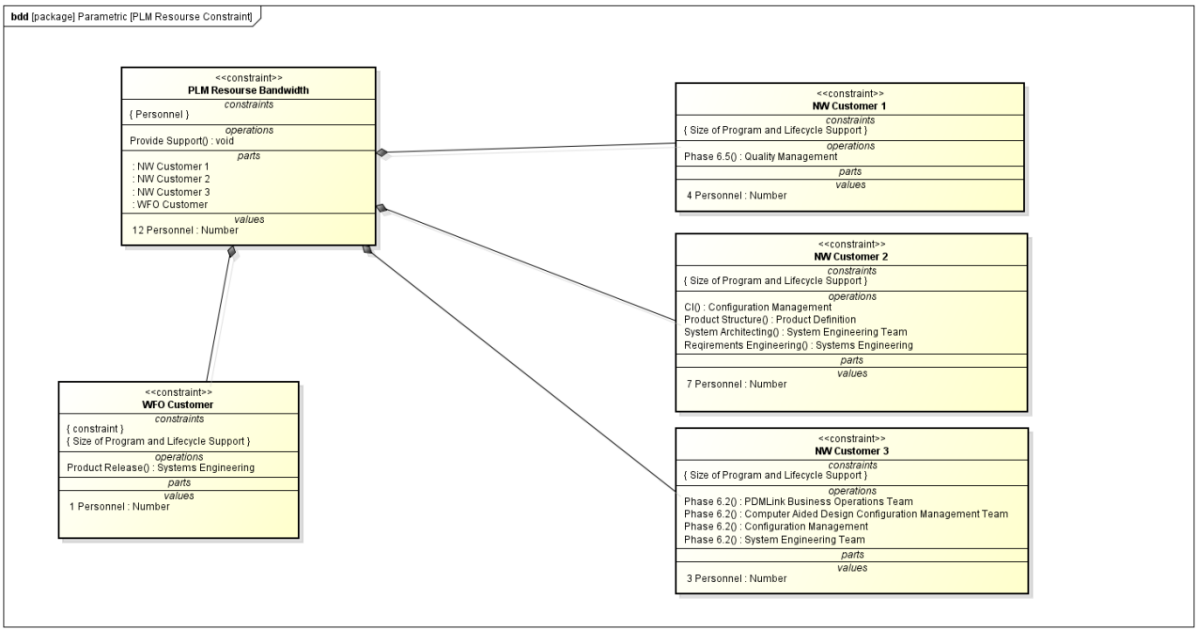


Figure 14. Block Definition Diagram Showing PLM Resource Constraints

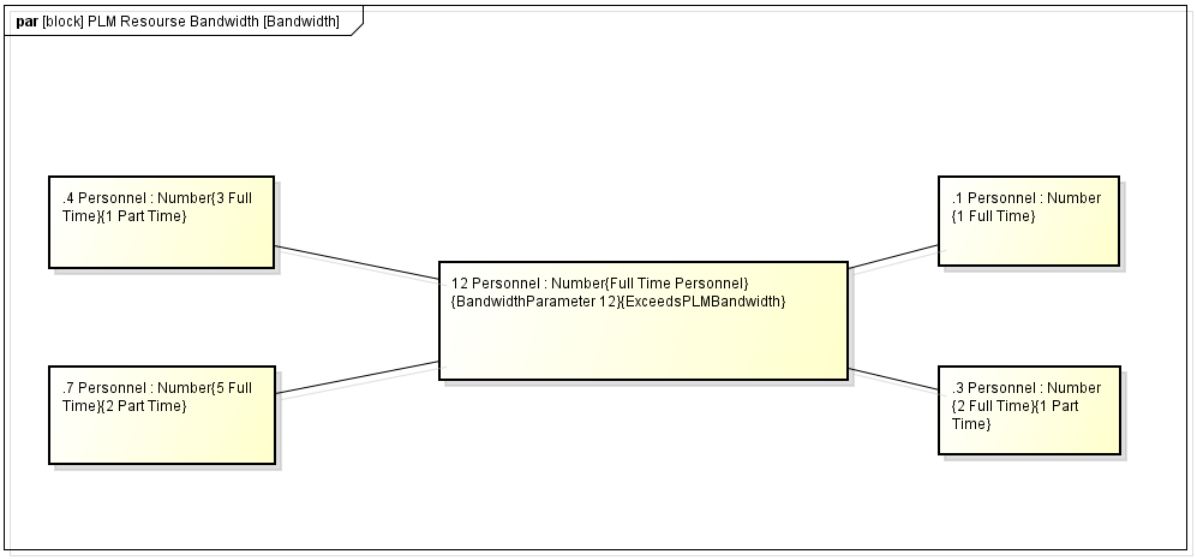


Figure 15. Parametric Diagram Showing PLM Resource Constraint

When displaying a parametric diagram, the main objective is to capture the constraints on a system. In this example, the constraints are represented as blocks in Figure 15 which shows the bindings between the block's value properties. It may not be necessary to display a parametric diagram for every block as Delligatti points out, "The power of this capability emerges when you apply the constraint expression to a block somewhere in your model to impose a fixed mathematical relationship on that block's value properties" (Delligatti, 2014). Creating

parametric diagrams can be powerful views of a model but first system architects must define the system ConOps.

5.2 Product Lifecycle Management Architecture Concept of Operations (ConOps)

For the purpose of this report, the PLM ConOps has been developed based on the need statement identified in Section 3.2 and the nuclear weapons program's 6.X process. Section 5.2.1 and Figure 16 detail the high level components of the 6.X. Section 5.2.2 details the PLM architecture and Figure 17 is an activity diagram of the PLM ConOps (dashed lines). As shown in Figure 17, the PLM architecture should be clearly traceable to the 6.X process based on originating requirements discussed in Section 4 (with the exception of phase 6.1). A few things were discovered during the development of the ConOps activity diagram for the PLM Architecture which are mentioned below.

- No clearly defined process for any type of PLM service exists or how to engage with the 6.X process exists
- The support level PLM should offer for initiation and planning of a nuclear weapons program is unknown
- Initiation and planning RPP's do not mention system architecting activities and functional/physical traceability to requirements

5.2.1 Nuclear Weapons Program ConOps

Nuclear Weapon Program 6.X
Built In Higher-Level Component(s):
DOM.1.3 Nuclear Weapon Program
Built From Lower-Level Component(s):

- 6.1 Nuclear Weapon Concept
- 6.2 Nuclear Weapon Feasibility & Cost Study
- 6.3 Nuclear Weapon Development Engineering
- 6.4 Nuclear Weapon Production Engineering
- 6.5 Nuclear Weapon First Production
- 6.6 Nuclear Weapon Quality Production

- 6.7 Nuclear Weapon Dismantlement

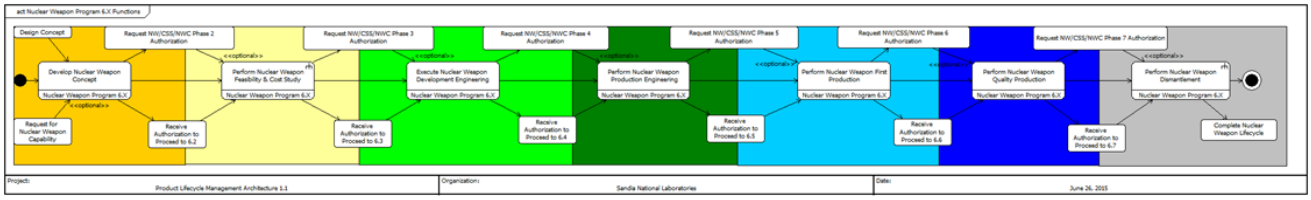


Figure 16. Activity Diagram of Nuclear Weapon 6.X Product Realization Lifecycle

5.2.2 Product Lifecycle Management ConOps

PLM.1 Product Lifecycle Management Architecture

PLM.1.1 Engage In Nuclear Weapon Concept Planning
PLM.1.2 Provide Product Lifecycle Management Plan
PLM.1.3 Implement Configuration Management Plan
PLM.1.4 Implement Systems Engineering Architecture
PLM.1.5 Implement Product Structure Architecture
PLM.1.6 Implement Requirements Engineering Architecture
PLM.1.7 Provide Maintenance & Support
PLM.1.8 Establish Software Support
PLM.1.9 Establish Creo Support
PLM.1.10 Establish Database Support
PLM.1.11 Determine Personnel Locations
PLM.1.12 Establish PDMLink Support
PLM.1.13 Perform Product Structure Functions
PLM.1.14 Perform Design Engineering Configuration Management Functions
PLM.1.15 Perform System Engineering Function
PLM.1.16 Perform Access Control Functions
PLM.1.17 Perform Product Definition & Tools Function
PLM.1.18 Complete Development Engineering Functions
PLM.1.19 Provide Production Product Structure Support
PLM.1.20 Provide Production Design Engineering Configuration Management Support
PLM.1.21 Provide Production System Engineering Support
PLM.1.22 Provide Production Product Definition Support
PLM.1.23 Complete Production Support
PLM.1.24 Provide 1st Production Product Definition Quality Control
PLM.1.25 Provide 1st Production Configuration Management Quality Control
PLM.1.26 Provide 1st Production Product Definition Maintenance Quality Control
PLM.1.27 Complete 1st Production Support
PLM.1.28 Provide Legacy Support

Built From Component(s):

PLM.1.1 Product Lifecycle Management Department
PLM.1.2 Configuration Management
PLM.1.3 Customer
PLM.1.4 Databases
PLM.1.5 Facilities

PLM.1.6 Product Definition
PLM.1.7 Software
PLM.1.8 Requirements Engineering
PLM.1.9 Systems Architecture

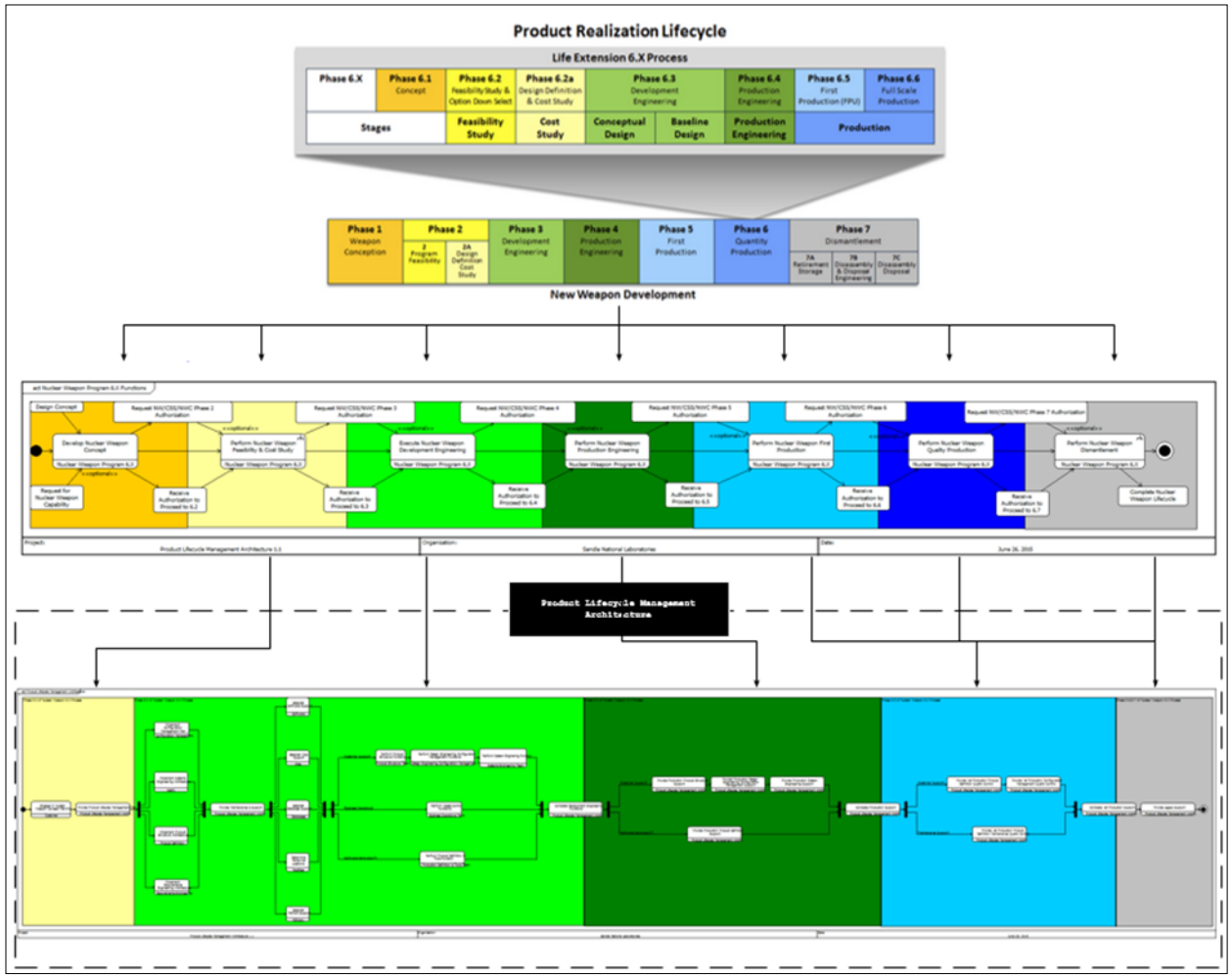


Figure 17. Activity Diagram of the Product Lifecycle Management & 6.X ConOps

5.2.3 Allocation of Product Lifecycle Management ConOps

This section contains details of the allocation of the PLM ConOps to components.

PLM.1.1 Engage In Nuclear Weapon Concept Planning

Allocated To:
PLM.1.3 Customer

PLM.1.2 Provide Product Lifecycle Management Plan

Allocated To:
PLM.1 Product Lifecycle Management Architecture Domain

PLM.1.3 Implement Configuration Management Plan

Allocated To:
PLM.1.2 Configuration Management

PLM.1.4 Implement Systems Engineering Architecture

Allocated To:
PLM.1.7.3 SysML

PLM.1.5 Implement Product Structure Architecture

Allocated To:
PLM.1.6 Product Definition

PLM.1.6 Implement Requirements Engineering Architecture

Allocated To:
PLM.1.8 Requirements Engineering

PLM.1.7 Provide Maintenance & Support

Allocated To:
PLM.1 Product Lifecycle Management Architecture Domain

PLM.1.8 Establish Software Support

Allocated To:
PLM.1.7 Software

PLM.1.9 Establish Creo Support

Allocated To:
PLM.1.7.1 Creo

PLM.1.10 Establish Database Support

Allocated To:
PLM.1.4 Databases

PLM.1.11 Determine Personnel Locations

Allocated To:
PLM.1.5 Facilities

PLM.1.12 Establish PDMLink Support

Allocated To:
PLM.1.7.2 PDMLink

PLM.1.13 Perform Product Structure Functions

Allocated To:
PLM.1.1.6 Product Structure Team

PLM.1.14 Perform Design Engineering Configuration Management Functions

Allocated To:
PLM.1.1.2 Design Engineering Configuration Management Team

PLM.1.15 Perform System Engineering Function

Allocated To:
PLM.1.1.4 Systems Engineering Team

PLM.1.16 Perform Access Control Functions

Allocated To:
PLM.1.1.1 Business Operations Team

PLM.1.17 Perform Product Definition & Tools Function

Allocated To:
PLM.1.1.3 Production Definition & Tools Team

PLM.1.18 Complete Development Engineering Functions

Allocated To:
PLM.1 Product Lifecycle Management Architecture Domain

PLM.1.19 Provide Production Product Structure Support

Allocated To:
PLM.1 Product Lifecycle Management Architecture Domain

PLM.1.20 Provide Production Design Engineering Configuration Management Support

Allocated To:
PLM.1 Product Lifecycle Management Architecture Domain

PLM.1.21 Provide Production System Engineering Support

Allocated To:
PLM.1 Product Lifecycle Management Architecture Domain

PLM.1.22 Provide Production Product Definition Support

Allocated To:
PLM.1 Product Lifecycle Management Architecture Domain

PLM.1.23 Complete Production Support

Allocated To:
PLM.1 Product Lifecycle Management Architecture Domain

PLM.1.24 Provide 1st Production Product Definition Quality Control

Allocated To:
PLM.1 Product Lifecycle Management Architecture Domain

PLM.1.25 Provide 1st Production Configuration Management Quality Control

Allocated To:
PLM.1 Product Lifecycle Management Architecture Domain

PLM.1.26 Provide 1st Production Product Definition Maintenance Quality Control

Allocated To:
PLM.1 Product Lifecycle Management Architecture Domain

PLM.1.27 Complete 1st Production Support

Allocated To:
PLM.1 Product Lifecycle Management Architecture Domain

PLM.1.28 Provide Legacy Support

Allocated To:
PLM.1 Product Lifecycle Management Architecture Domain

5.3 Model of the Product Lifecycle Management Domain

Based on the ConOps in 5.2.2, a block definition of the PLM architecture domain is listed below including the 6.X process which is shown in Figure 18.

DOM.1 Product Lifecycle Management Domain
- DOM.1.1 Customer Databases
- DOM.1.2 Software Tools
  - DOM.1.2.1 DPNet4
  - DOM.1.2.2 Engineering Web
  - DOM.1.2.3 FileNet Share
  - DOM.1.2.4 Image Management System
  - DOM.1.2.5 SharePoint
  - PLM.1.7.1 Creo
  - PLM.1.7.2 PDMLink
  - PLM.1.7.3 SysML
- DOM.1.3 Nuclear Weapon Program
  - 6 Nuclear Weapon Programs 6.X
    - 6.1 Nuclear Weapon Concept
    - 6.2 Nuclear Weapon Feasibility & Cost Study
      - 6.2.1 Design Definition Cost Study
      - 6.2.2 Program Feasibility
    - 6.3 Nuclear Weapon Development Engineering
    - 6.4 Nuclear Weapon Production Engineering
    - 6.5 Nuclear Weapon First Production
    - 6.6 Nuclear Weapon Quality Production

6.7 Nuclear Weapon Dismantlement

6.7.1 Disassembly & Disposal Engineering

6.7.2 Disassembly Disposal

6.7.3 Retirement Storage

DOM.1.4 Regulatory Agencies

DOM.1.4.1 Department of Defense

DOM.1.4.2 Department of Energy

DOM.1.4.3 National Nuclear Security Administration

DOM.1.4.4 Sandia National Laboratories

DOM.1.6 Work for Others

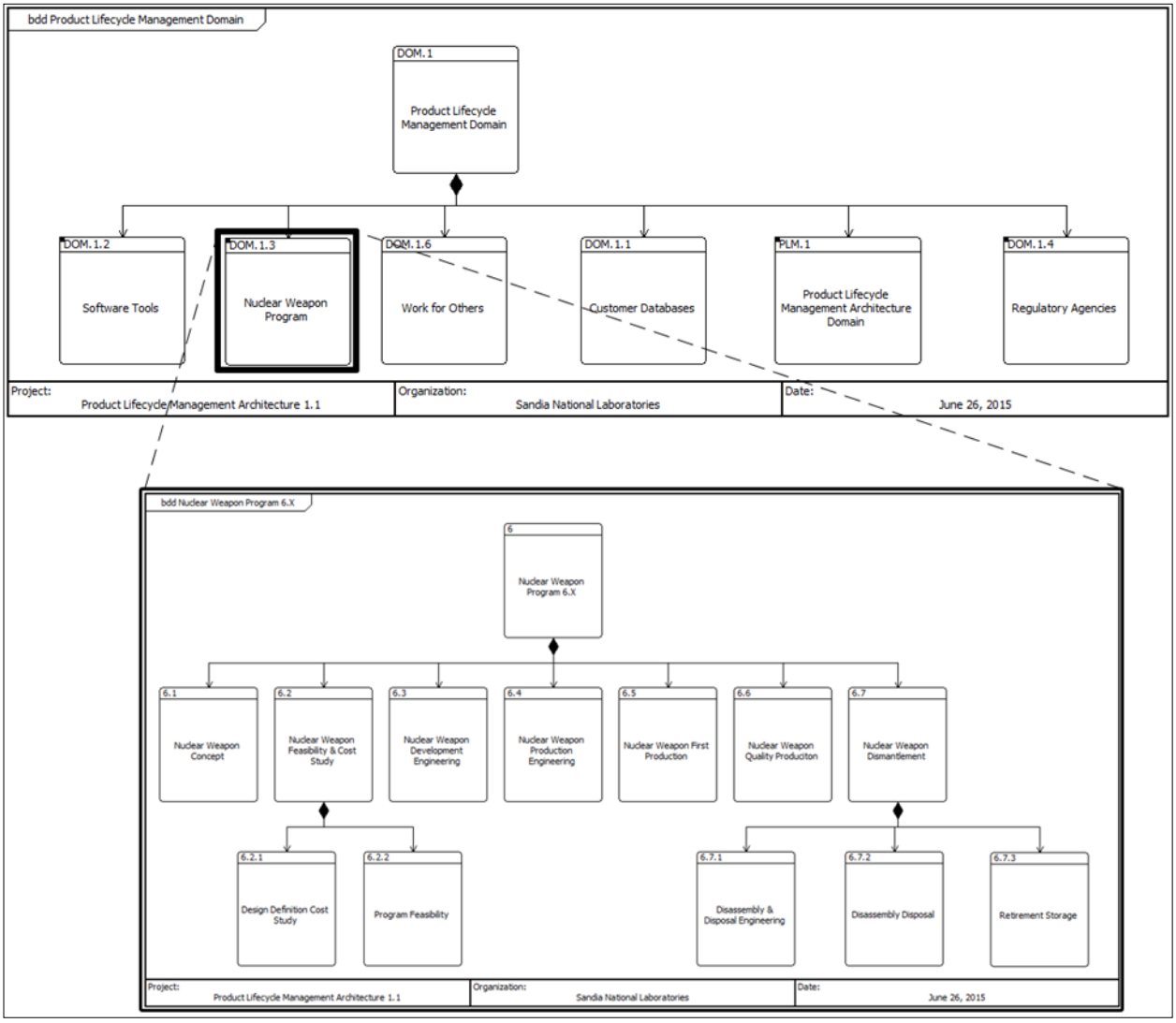


Figure 18. Block Definition of the Product Lifecycle Management Domain & 6.X Process

5.4 Model of the Product Lifecycle Management Architecture Domain

Below are the details for the domain of the PLM architecture which are shown in a series of domain diagrams (Figures 19-20).

PLM.1 Product Lifecycle Management Architecture (Level 1)

Description:
Level 1 of the Product Lifecycle Management Architecture Domain.

Type: System Architecture

Built In Higher-Level Component(s):
DOM.1 Product Lifecycle Management Domain

Built From Lower-Level Component(s):
PLM.1.1 Product Lifecycle Management Department
PLM.1.2 Configuration Management
PLM.1.3 Customer
PLM.1.4 Databases
PLM.1.5 Facilities
PLM.1.6 Product Definition
PLM.1.7 Software
PLM.1.8 Requirements Engineering
PLM.1.9 Systems Architecture

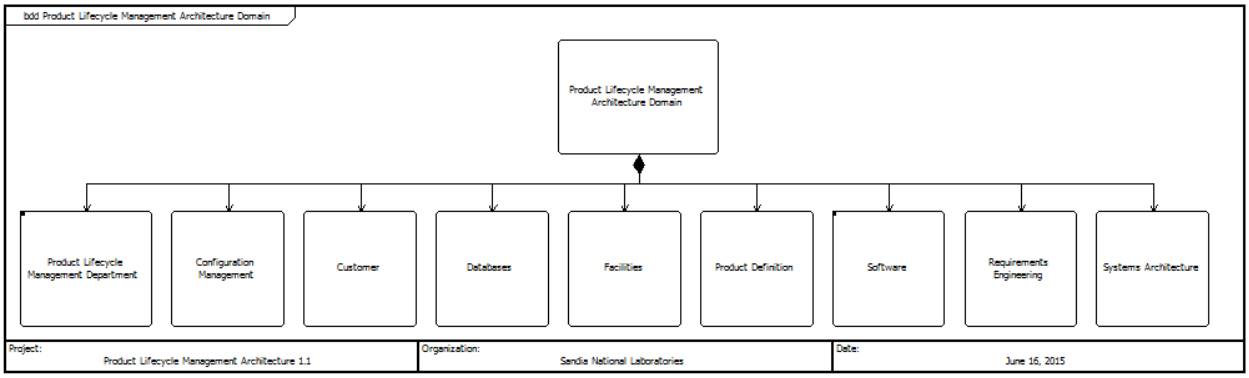


Figure 19. Product Lifecycle Management Architecture Domain Diagram Level 1

PLM.1.1 Product Lifecycle Management Department (Level 2)

Built In Higher-Level Component(s):
PLM.1 Product Lifecycle Management Architecture Domain

Built From Lower-Level Component(s):
PLM.1.1.1 Business Operations Team
PLM.1.1.2 Design Engineering Configuration Management Team
PLM.1.1.3 Production Definition & Tools Team
PLM.1.1.4 Systems Engineering Team
PLM.1.1.5 PDMLink Business Team
PLM.1.1.6 Product Structure Team
PLM.1.1.7 Software Tool Maintenance

PLM.1.1.8 Product Lifecycle Management Personnel

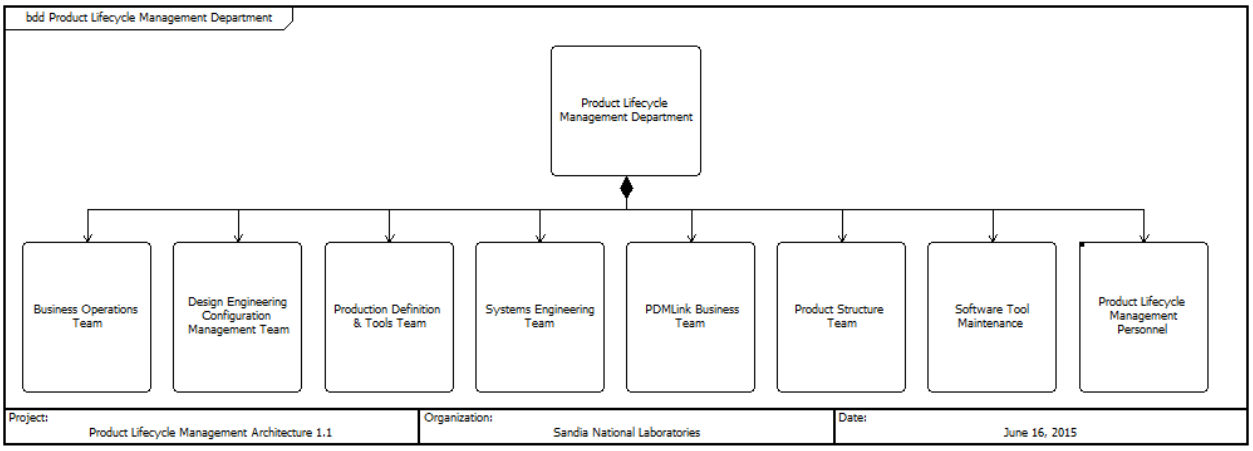


Figure 20. Product Lifecycle Management Architecture Domain Diagram Level 2

PLM.1.1.8 Product Lifecycle Management Personnel (Level 3)

Built In Higher-Level Component(s):

PLM.1.1 Product Lifecycle Management Department

Built From Lower-Level Component(s):

PLM.1.1.8.1 Computer Aided Design & Drafting Technologist
PLM.1.1.8.2 Engineering Document Control Technologist
PLM.1.1.8.3 Engineering Program/Project Lead
PLM.1.1.8.4 Engineering Systems Integration/Implementation Pro
PLM.1.1.8.5 Information Systems Security Technologist
PLM.1.1.8.6 Manager, R&D Science & Engineering
PLM.1.1.8.7 R&D S&E, Mechanical Engineering
PLM.1.1.8.8 Solutions Architect
PLM.1.1.8.9 Systems Engineer

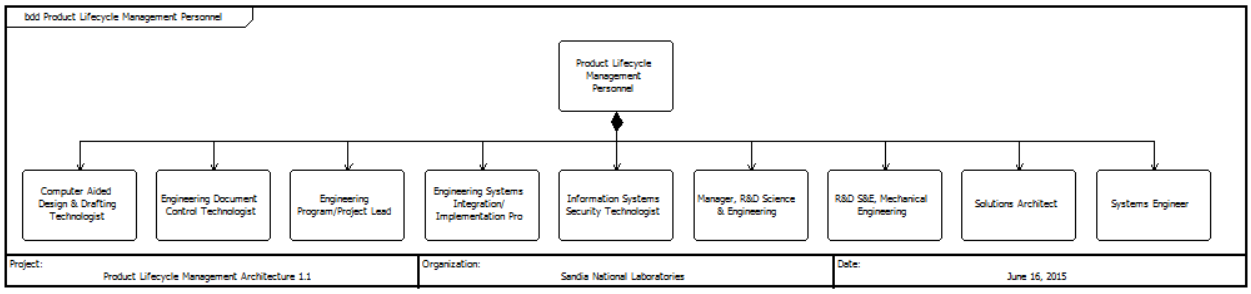


Figure 21. Product Lifecycle Management Architecture Domain Diagram Level 3

6 SOLUTION SPACE: DEVELOPING REQUIREMENTS

Based on the RPP requirements discussed in Section 4 and the ConOps in Section 5, "system" level requirements specific to the PLM have been developed through the practice of use cases and sequence diagrams. As described by Delligatti, "A use case diagram is a black-box view of the system" (Delligatti, 2014). Delligattie further explains, "A use case diagram is an analysis tool and is generally created early in the system life cycle. System analysis may enumerate use cases and create use case diagrams during the development of the system concept of operations (ConOps)" (Delligatti, 2014). Three levels of the PLM Architecture are analyzed using use cases in relation to the ConOps and the 6.X phases. The result of the analysis is a series of requirements which are represented in a requirements hierarchy after each series of use cases and sequence diagrams. The details of these requirements are captured in Section 7 and a Requirements Traceability Matrix in Appendix A.

6.1 Product Lifecycle Management Architecture and Nuclear Weapon Phase 6.2

This use-case describes the PLM Architecture in relation to phase 6.2.

- Level 1 use case for NW 6.2: Product Lifecycle Management Architecture Initiation & Planning. Discussions and planning between the PLM manager and appropriate Nuclear Weapon Strategic Management Unit personnel regarding the nuclear weapon concept and PLM support.

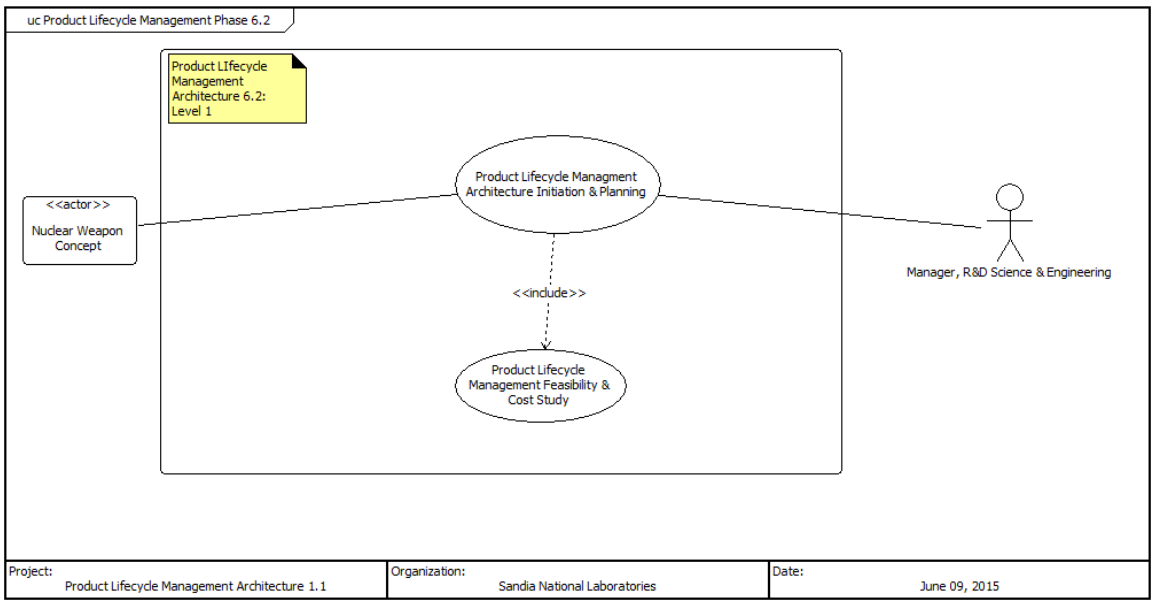


Figure 22. Use Case: Product Lifecycle Management & Nuclear Weapon 6.2 Level 1

- Level 2 use case for NW 6.2: Product Lifecycle Management Feasibility & Cost Study. Based on the planning in the previous scenario, the PLM is directed to produce plans for product structure, system architecting, configuration management, and requirements engineering.

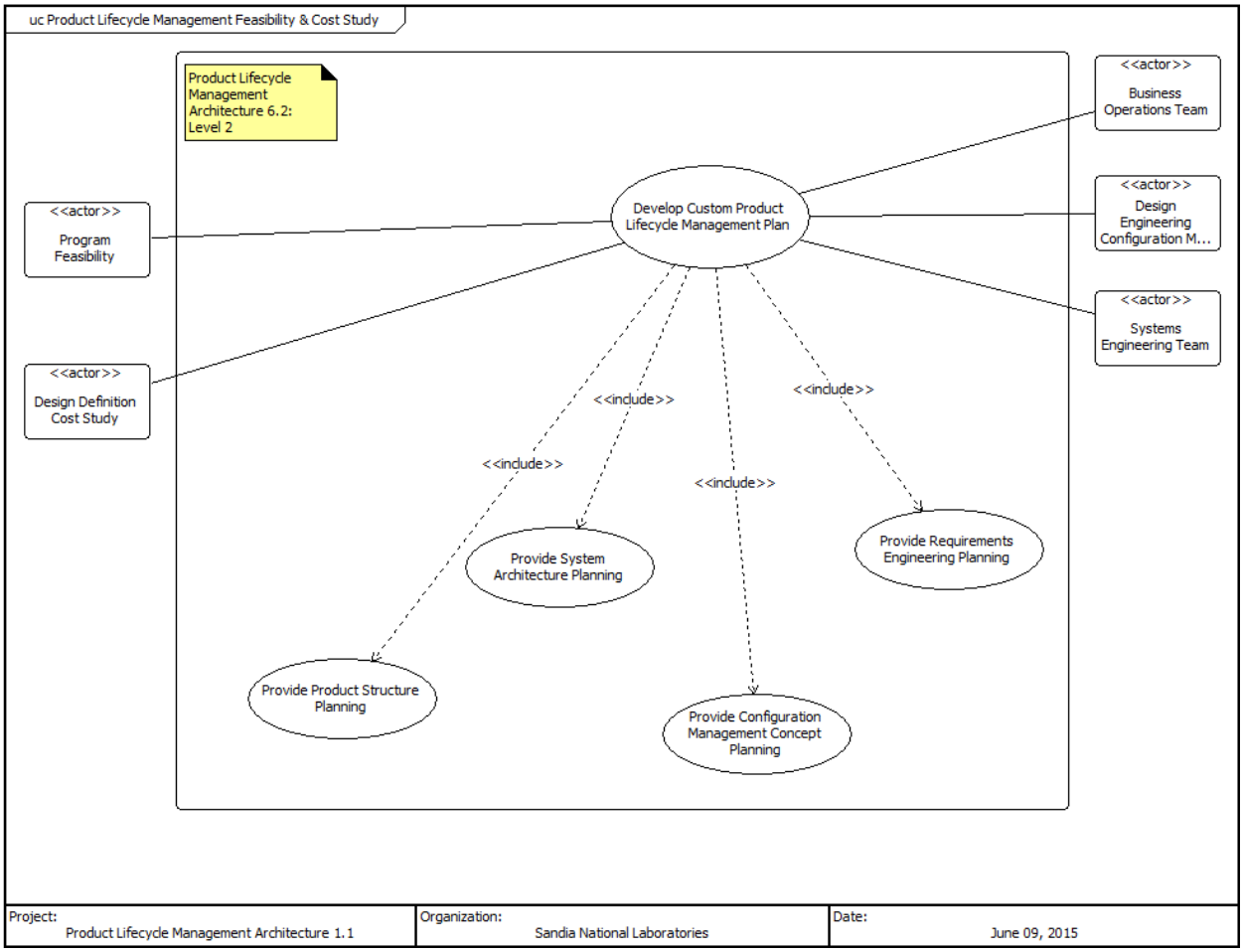


Figure 23. Use Case: Product Lifecycle Management & Nuclear Weapon 6.2 Level 2

Level 3 contains four use cases for NW phase 6.2.

- Level 3 use case 3.1 for NW 6.2: Provide Product Structure Planning. Each plan will be developed by the appropriate PLM personnel with input from the customer.

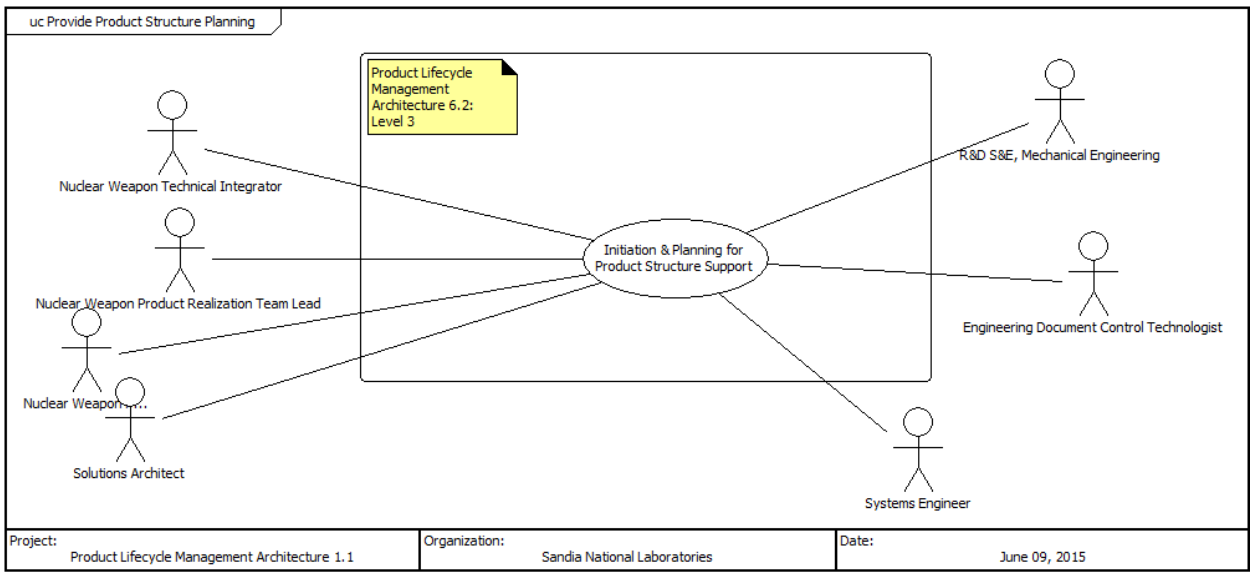


Figure 24. Use Case 3.1: Product Lifecycle Management & Nuclear Weapon 6.2 Level 3

- Level 3 use case 3.2 for NW 6.2: Provide System Architecture Planning

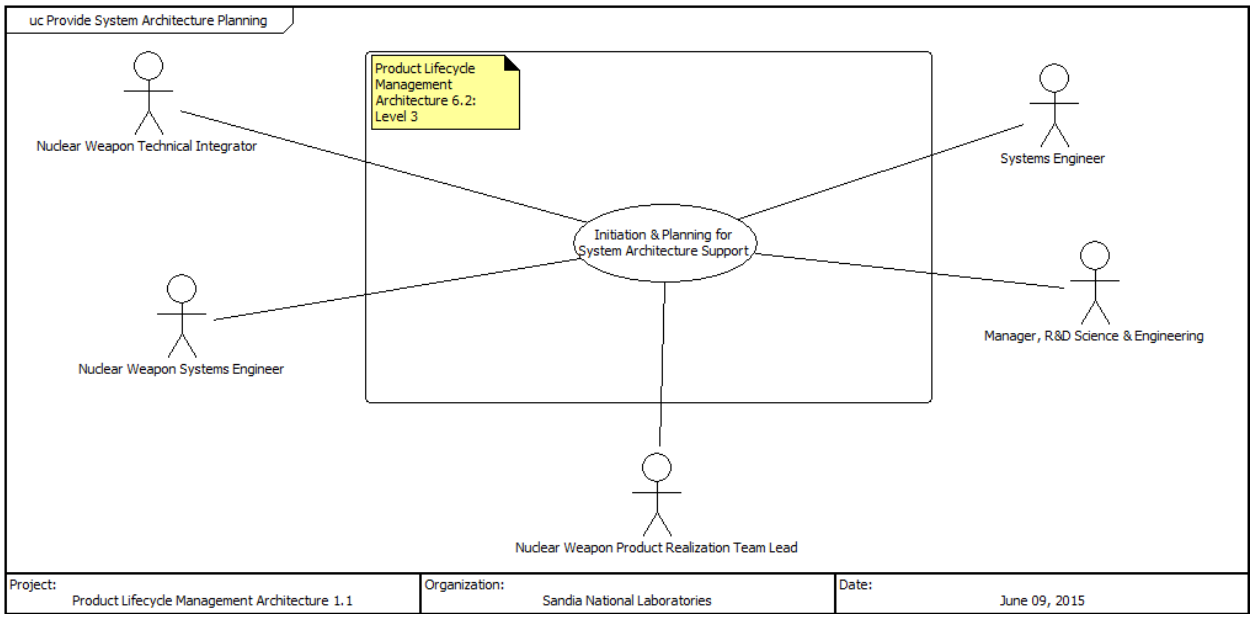


Figure 25. Use Case 3.2: Product Lifecycle Management & Nuclear Weapon 6.2 Level 3

- Level 3 use case 3.3 for NW 6.2 :Provide Configuration Management Concept Planning

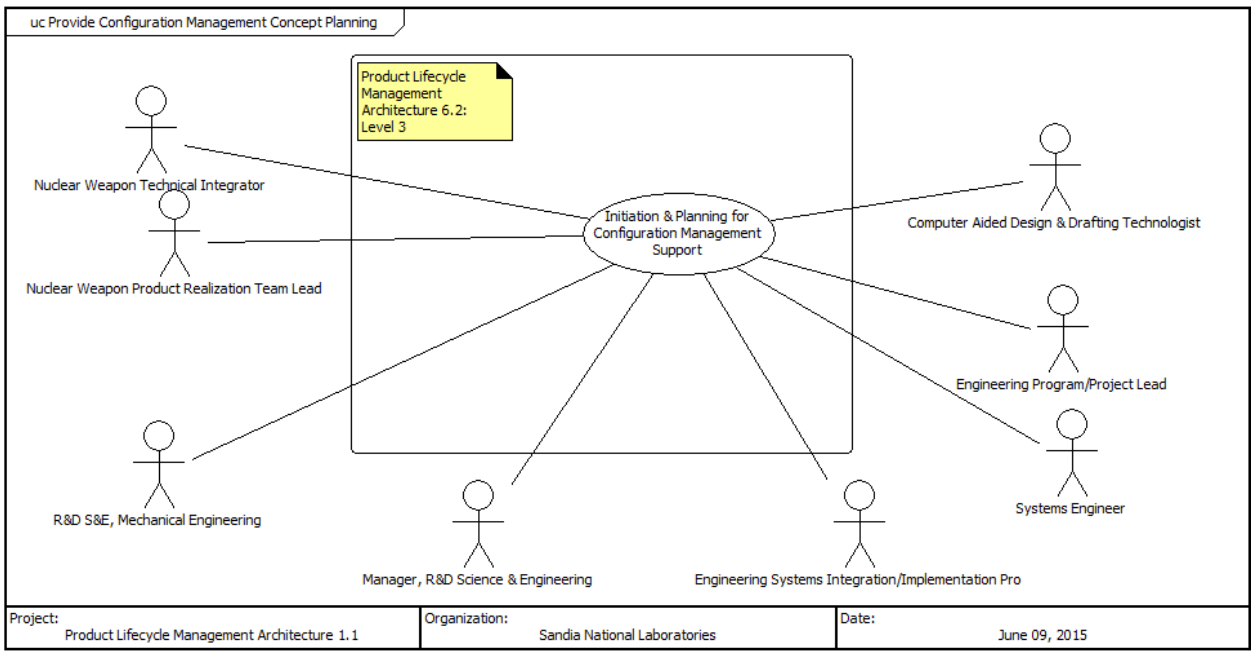


Figure 26. Use Case 3.3: Product Lifecycle Management & Nuclear Weapon 6.2 Level 3

- Level 3 use case 3.4 for NW 6.2: Provide Requirements Engineering & Management Planning

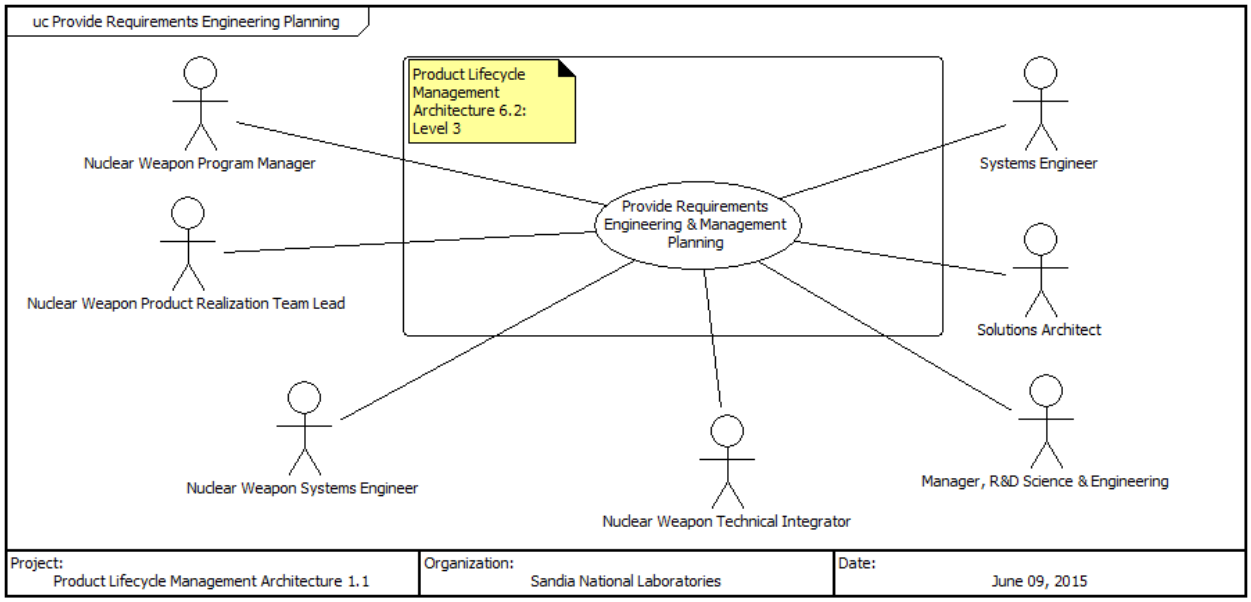


Figure 27. Use Case 3.4: Product Lifecycle Management & Nuclear Weapon 6.2 Level 3

Based on the use-cases for PLM and phase 6.2, a sequence diagram was created to show relation to high-level requirements developed for the architecture.

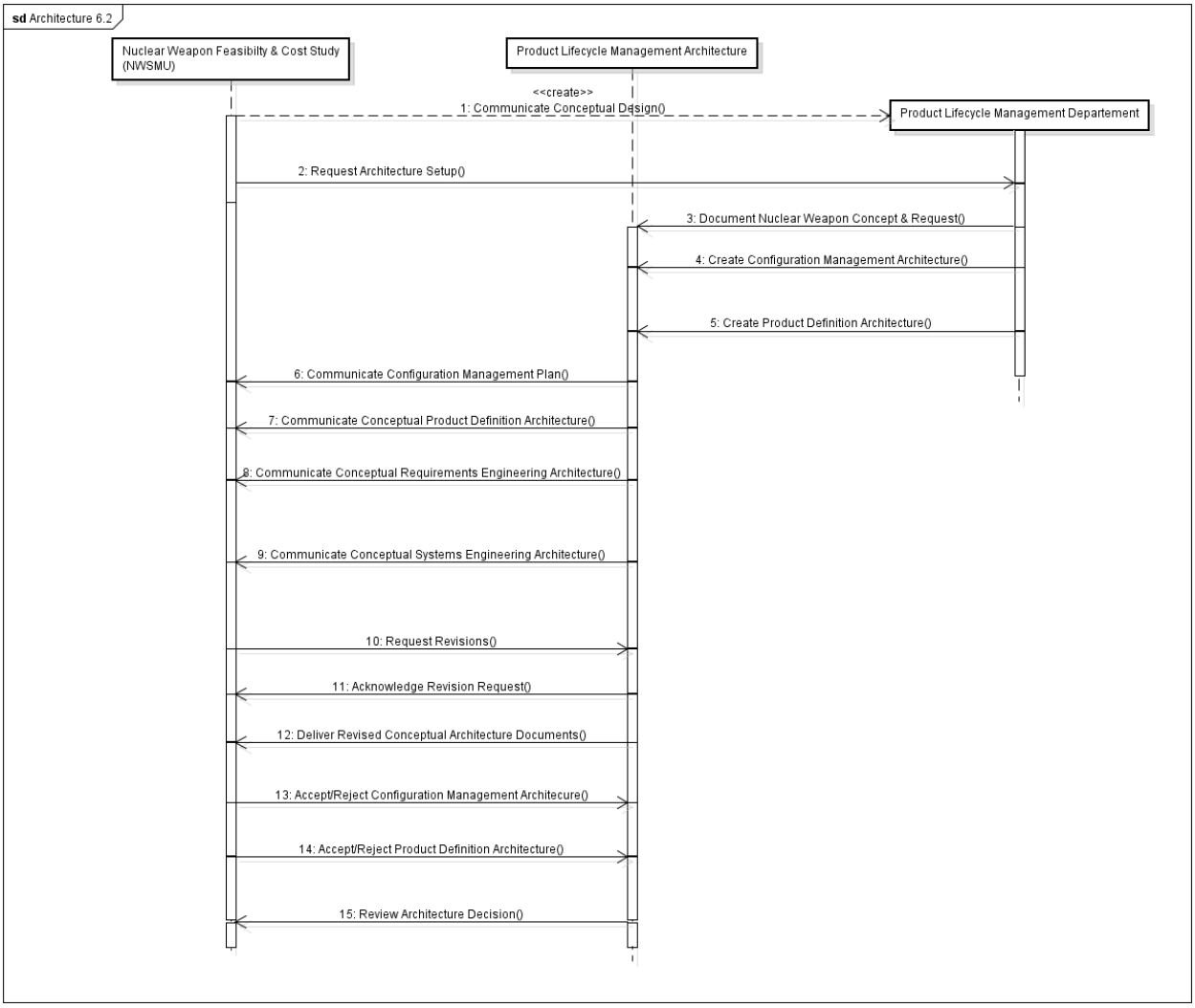


Figure 28. Product Lifecycle Management Architecture Sequence Diagram for Phase 6.2

Based on the above methodology, an initiation set of requirements were developed and are shown in Figure 29. The abbreviation for this set is INT.

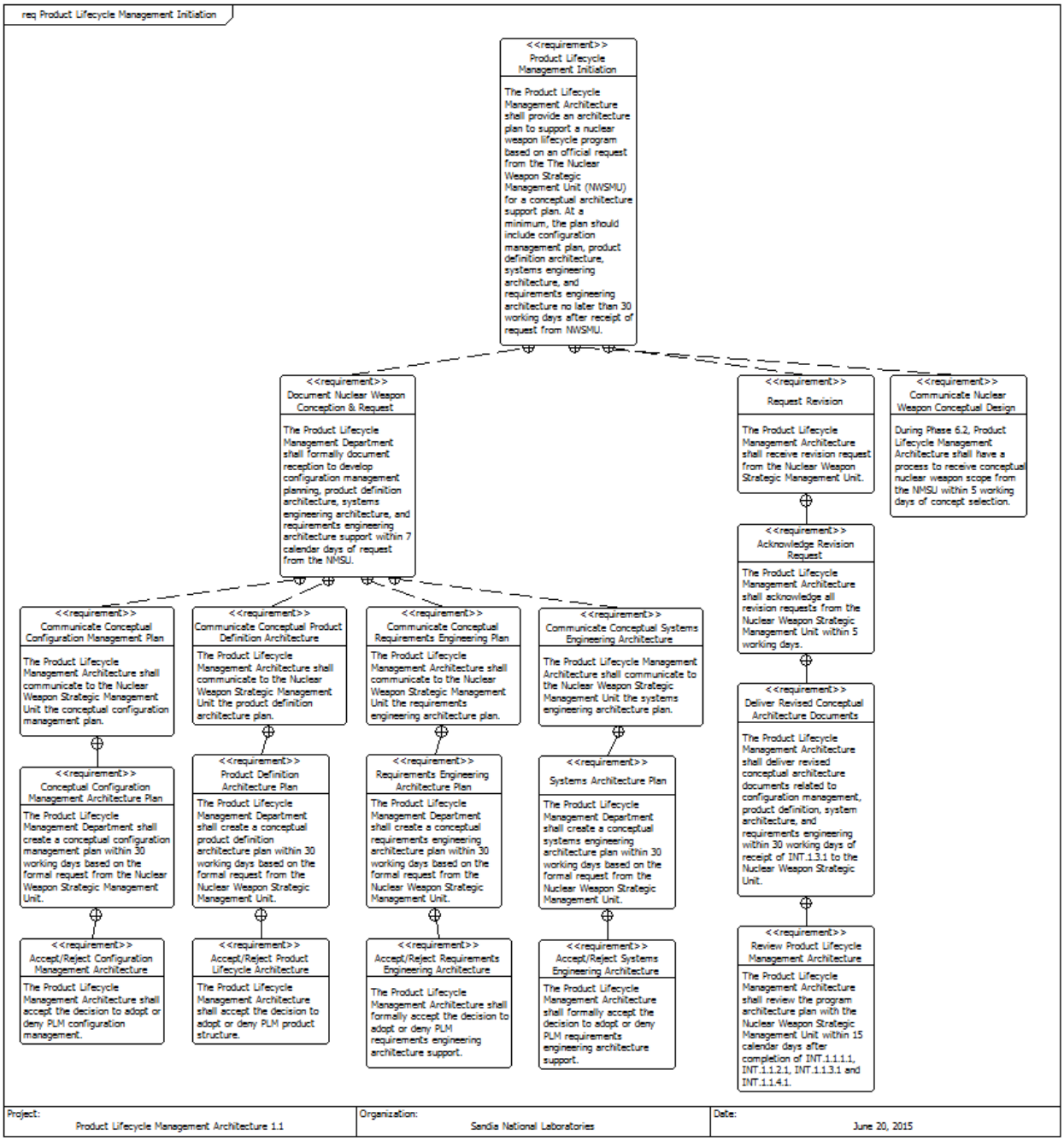


Figure 29. Product Lifecycle Management Initiation Set of Requirements

6.2 Product Lifecycle Management Architecture and Nuclear Weapon Phase 6.3

This use case describes the PLM Architecture in relation to phase 6.3.

- Level 1 use case for NW 6.3: Product Lifecycle Management Architecture Execution. This series of use cases describes how the PLM will execute the architecture once the planning has been accepted by the customer.

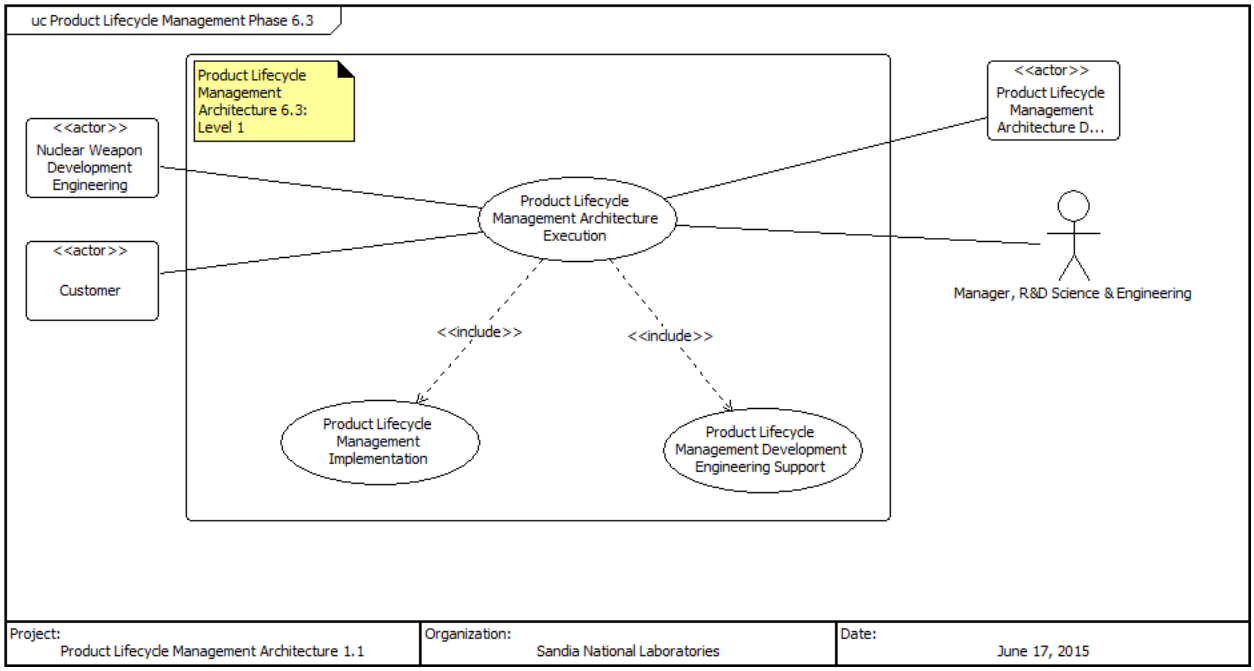


Figure 30. Use Case: Product Lifecycle Management & Nuclear Weapon 6.3 Level 1

Level 2 contains two use cases for NW phase 6.3.

- Level 2.1 use case for NW 6.3: Product Lifecycle Management Implementation. This use case describes how the PLM will implement the four main services and the appropriate actors.

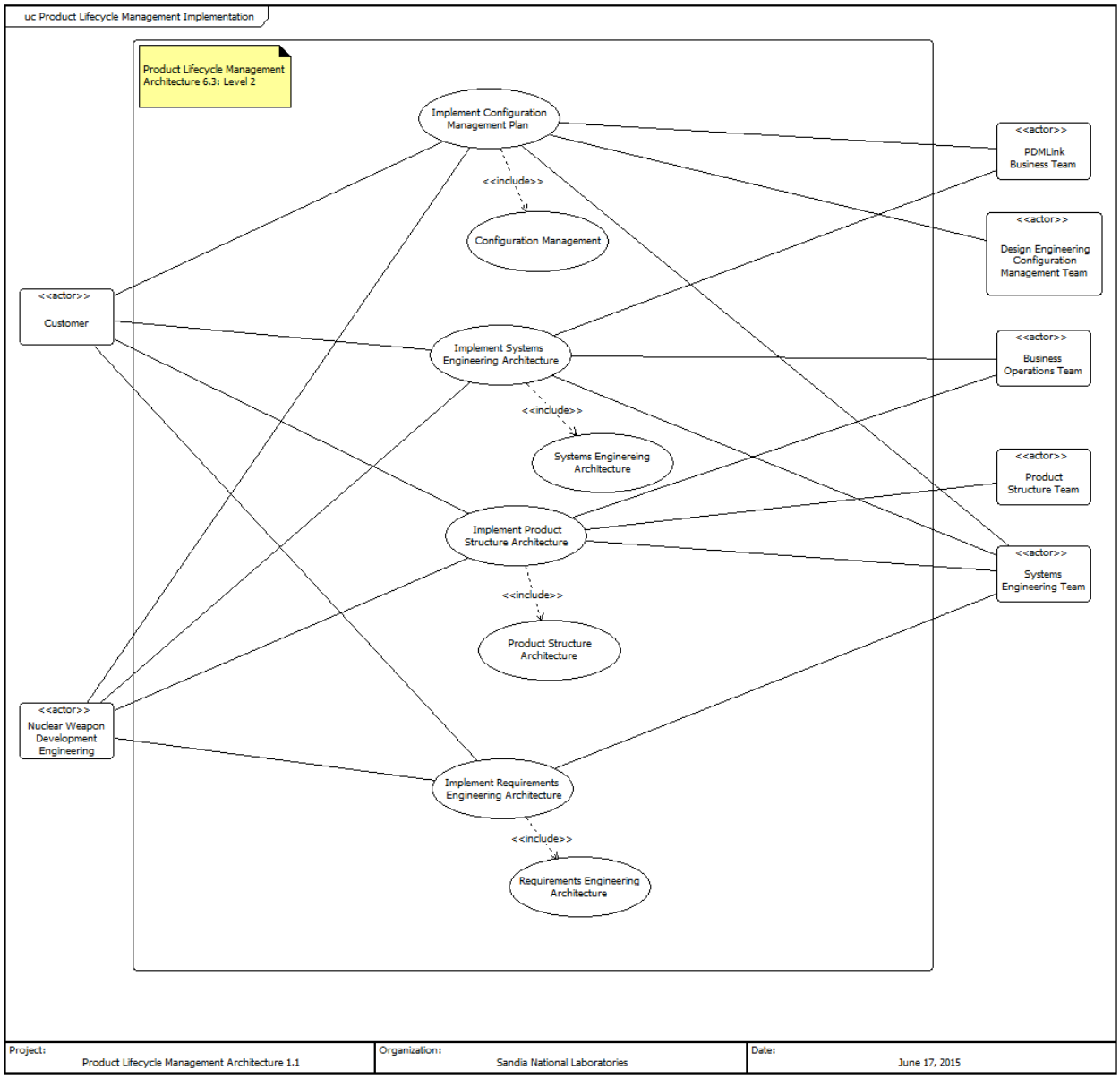


Figure 31. Use Case 2.1: Product Lifecycle Management & Nuclear Weapon 6.3 Level 2

- Level 2 use case 2.2 for NW 6.3: Product Lifecycle Management Development Engineering Support. This use case describes how the PLM will provide support to the customer during the development engineering phase.

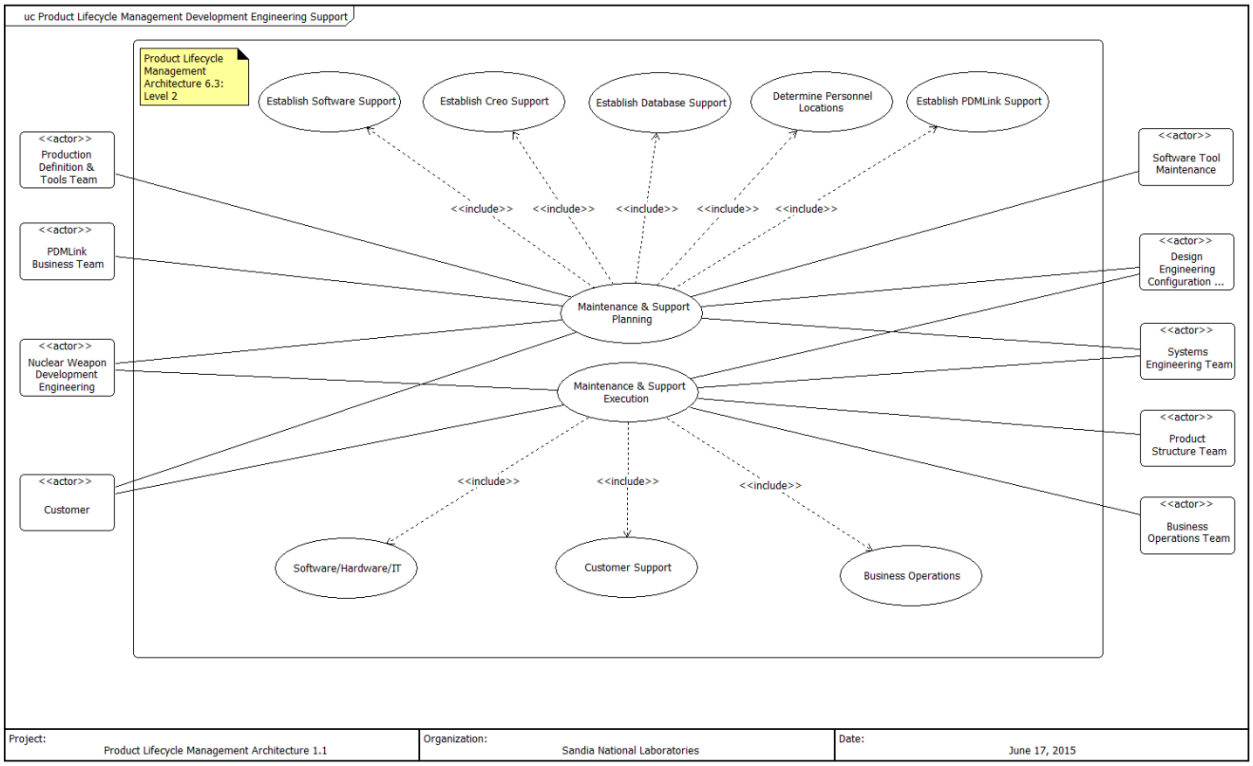


Figure 32. Use Case 2.2: Product Lifecycle Management & Nuclear Weapon 6.3 Level 2

Level 3 contains twelve use cases for NW phase 6.3.

- Level 3 use case 3.1 for NW 6.3: Configuration Management. This use case describes the actors involved in configuration management during the 6.3 phase. At this level, the use case describes the personnel involved from the customer and the PLM.

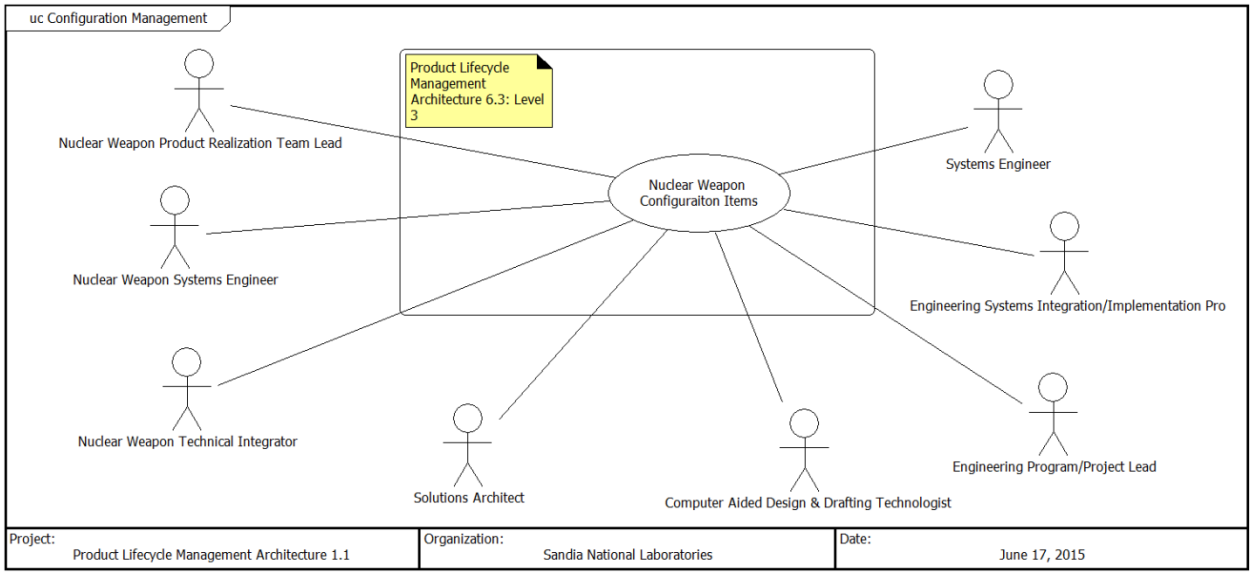


Figure 33. Use Case 3.1: Product Lifecycle Management & Nuclear Weapon 6.3 Level 3

- Level 3 use case 3.2 for NW 6.3: System Engineering Architecture. This use case describes the actors involved in system engineering during the 6.3 phase. At this level, the use case describes the personnel involved from the customer and the PLM.

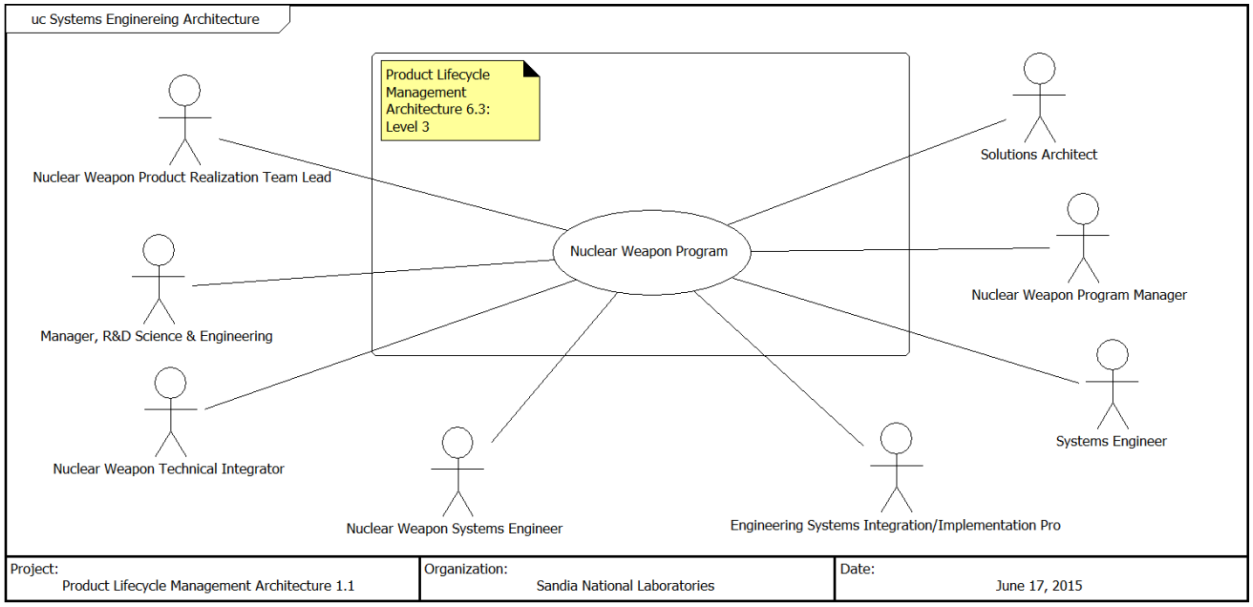


Figure 34. Use Case 3.2: Product Lifecycle Management & Nuclear Weapon 6.3 Level 3

- Level 3 use case 3.3 for NW 6.3: Product Structure Architecture. This use case describes the actors involved in product structure during the 6.3 phase. At this level, the use case describes the personnel involved from the customer and the PLM.

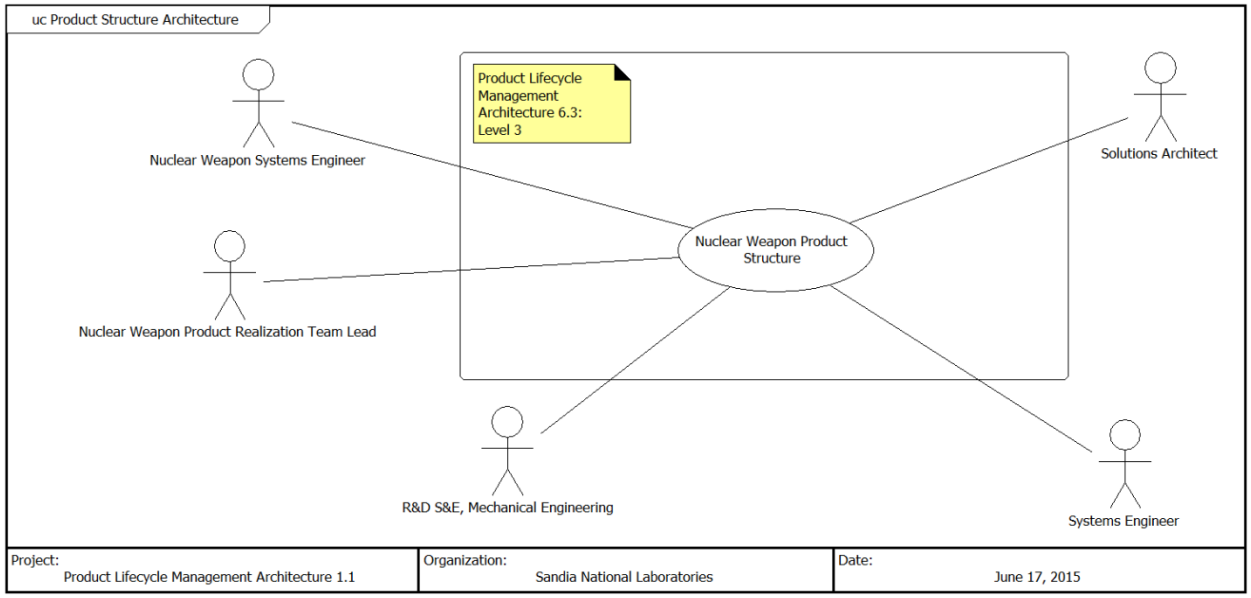


Figure 35. Use Case 3.3: Product Lifecycle Management & Nuclear Weapon 6.3 Level 3

- Level 3 use case 3.4 for NW 6.3: Requirements Engineering Architecture. This use case describes the actors involved in requirements engineering during the 6.3 phase. At this level, the use case describes the personnel involved from the customer and the PLM.

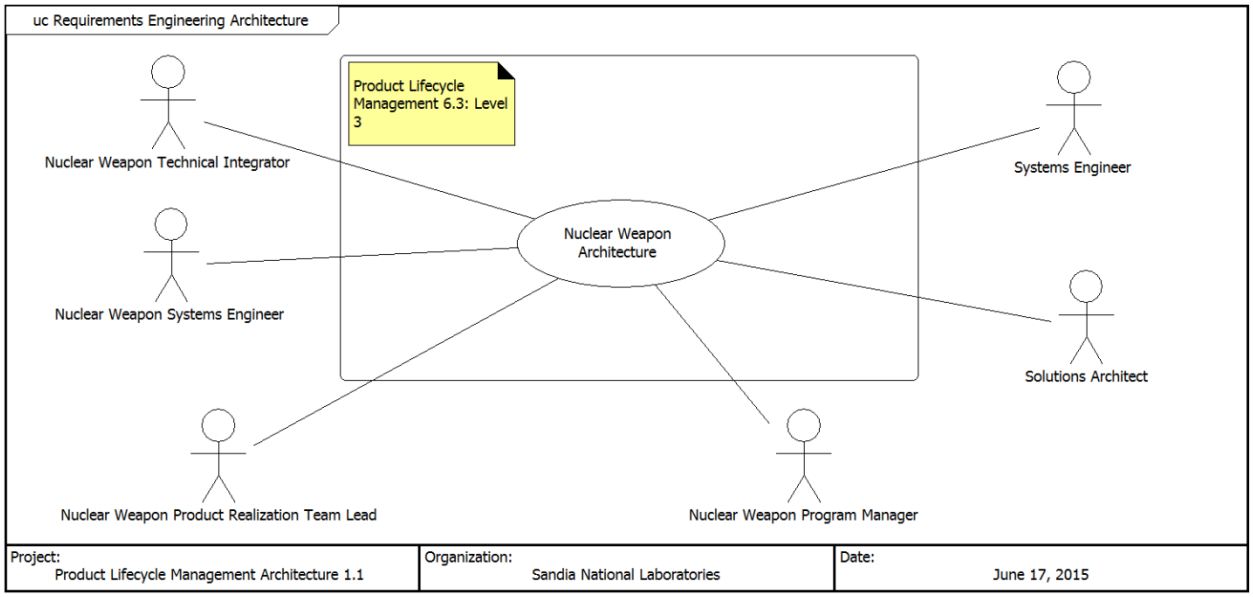


Figure 36. Use Case 3.4: Product Lifecycle Management & Nuclear Weapon 6.3 Level 3

- Level 3 use case 3.5 for NW 6.3: Establish Software Support. This use case describes the actors involved in software support during the 6.3 phase.

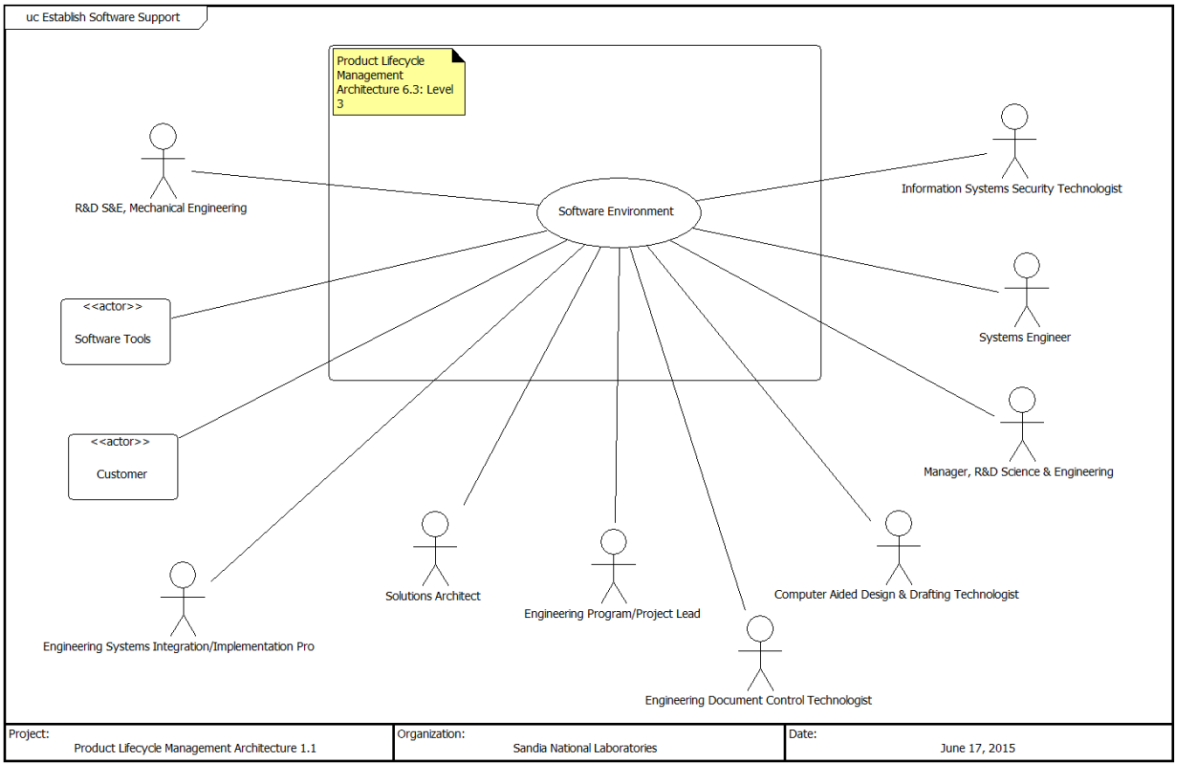


Figure 37. Use Case 3.5: Product Lifecycle Management & Nuclear Weapon 6.3 Level 3

- Level 3 use case 3.6 for NW 6.3: Establish Creo Support. This use case describes the actors involved in Creo support during the 6.3 phase. The difference between providing Creo support and additional software support is that Creo is a known product the PLM works with. Based on the services each customer desires, there may be additional software support.

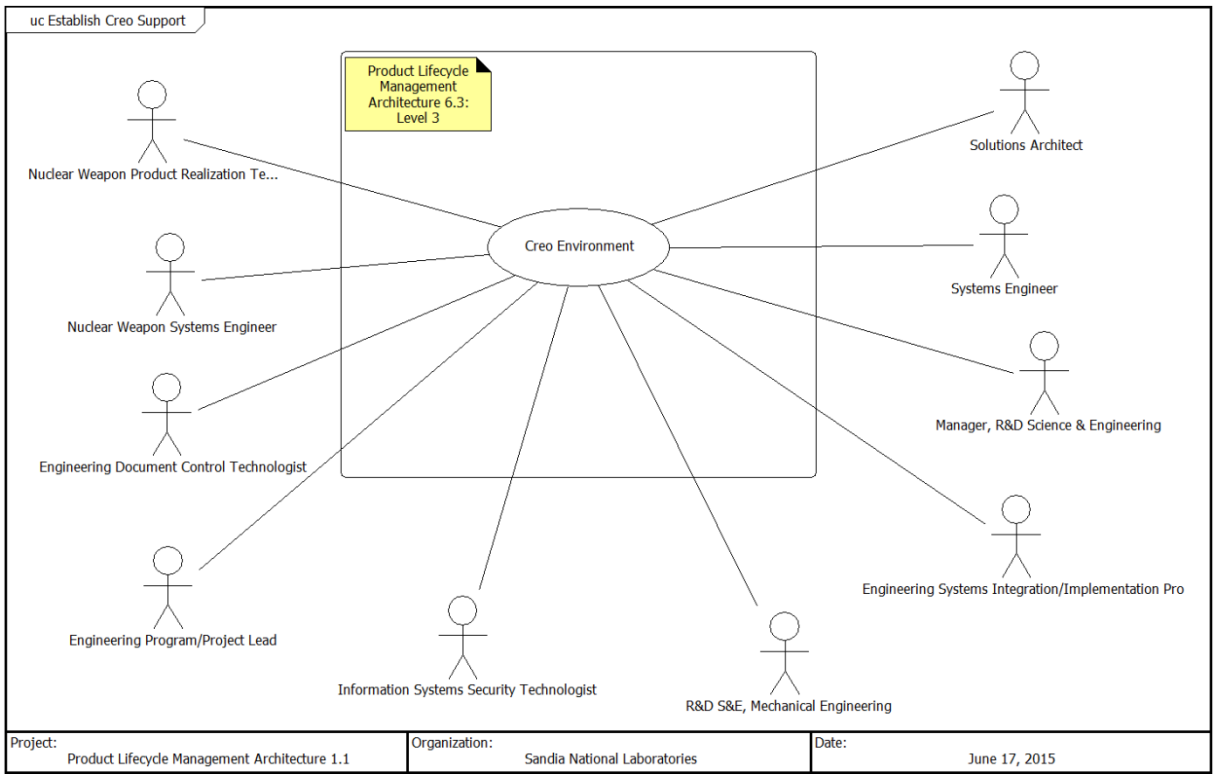


Figure 38. Use Case 3.6: Product Lifecycle Management & Nuclear Weapon 6.3 Level 3

- Level 3 use case 3.7 for NW 6.3: Establish Database Support. This use case describes the actors involved in support for databases during the 6.3 phase.

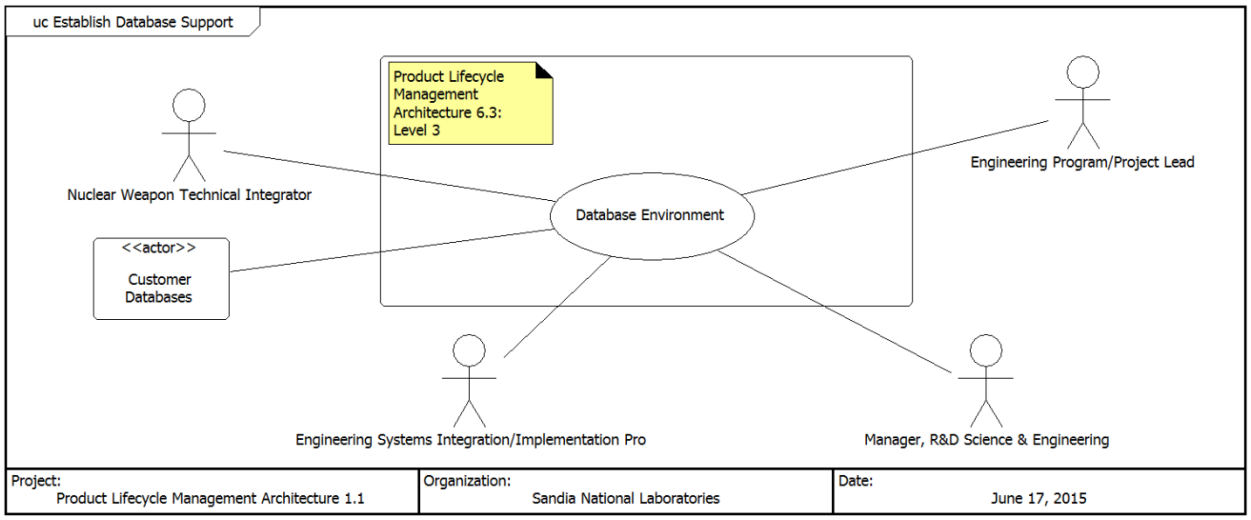


Figure 39. Use Case 3.7: Product Lifecycle Management & Nuclear Weapon 6.3 Level 3

- Level 3 use case 3.8 for NW 6.3: Determine Personal Locations. This use case describes the actors involved in determining where deployed personnel will reside during the 6.3 phase.

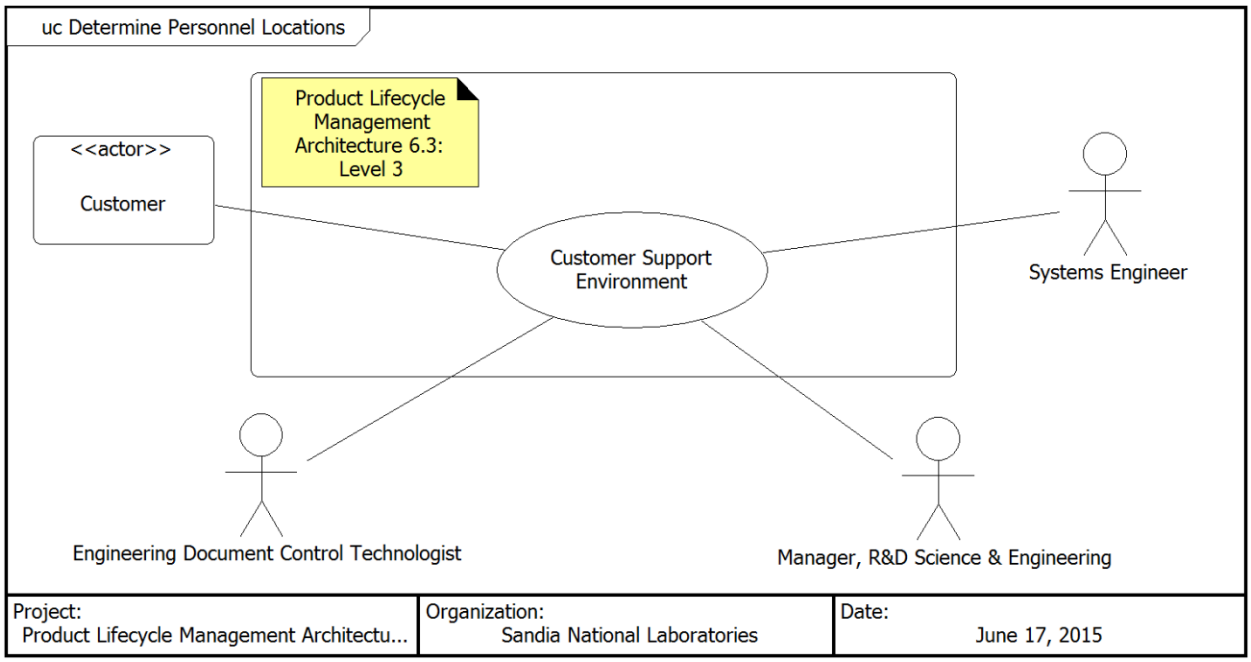


Figure 40. Use Case 3.8: Product Lifecycle Management & Nuclear Weapon 6.3 Level 3

- Level 3 use case 3.9 for NW 6.3: Establish PDMLink Support. This use case describes the actors involved in providing PDMLink support to the customer during the 6.3 phase.

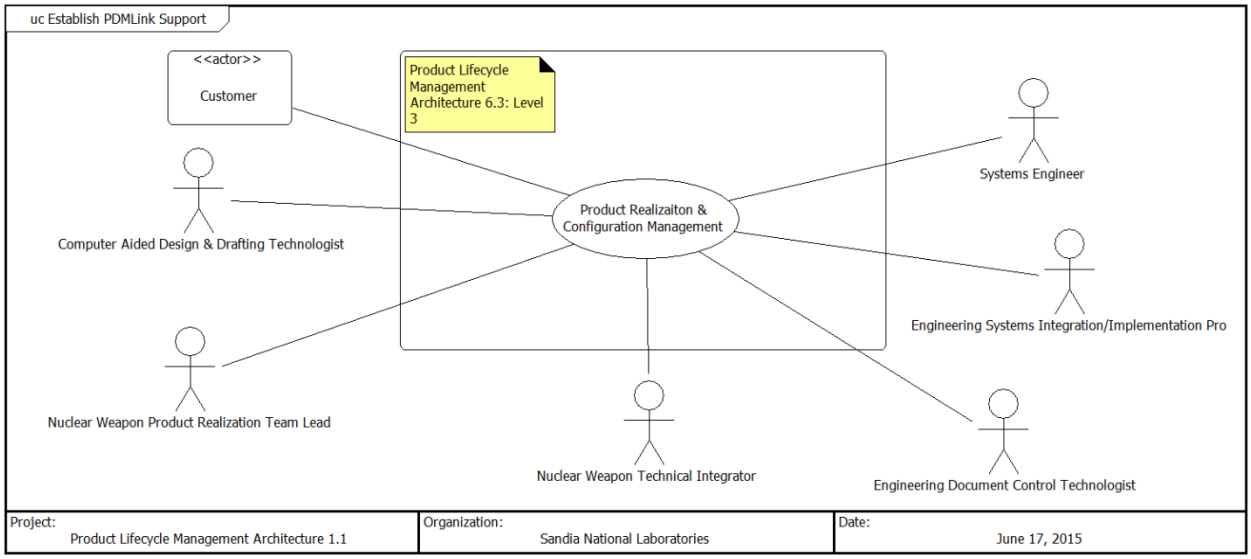


Figure 41. Use Case 3.9: Product Lifecycle Management & Nuclear Weapon 6.3 Level 3

- Level 3 use-case 3.10 for NW 6.3: Software/Hardware/IT. This use case describes the actors involved in additional software/hardware/IT support needed through help tickets during the 6.3 phase.

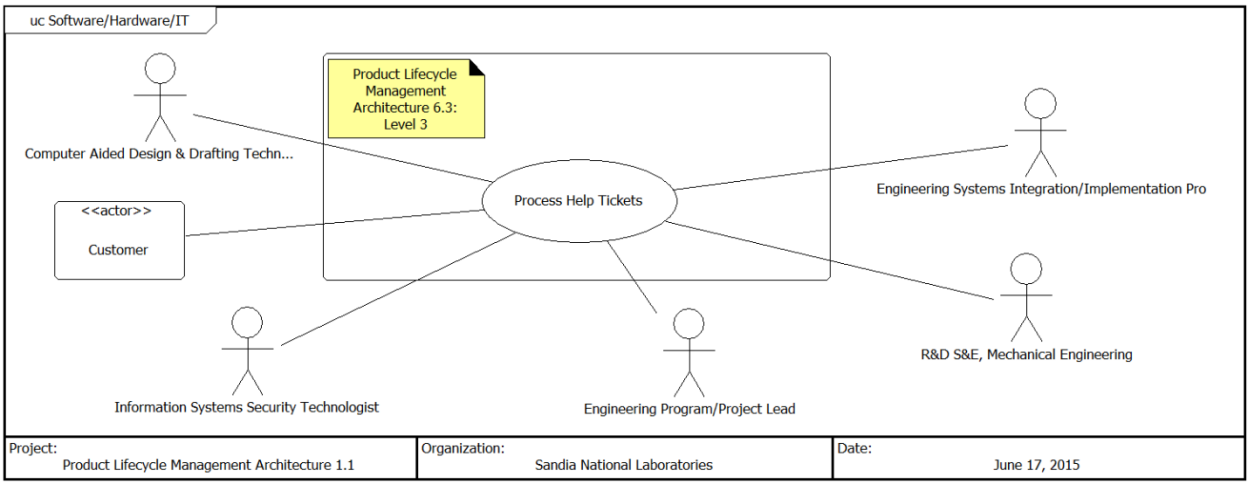


Figure 42. Use Case 3.10: Product Lifecycle Management & Nuclear Weapon 6.3 Level 3

- Level 3 use case 3.11 for NW 6.3: Customer Support. This use case describes the actors involved in customer support regarding configuration management and systems engineering during the 6.3 phase. At this level, the use case describes the personnel involved from the customer and the PLM

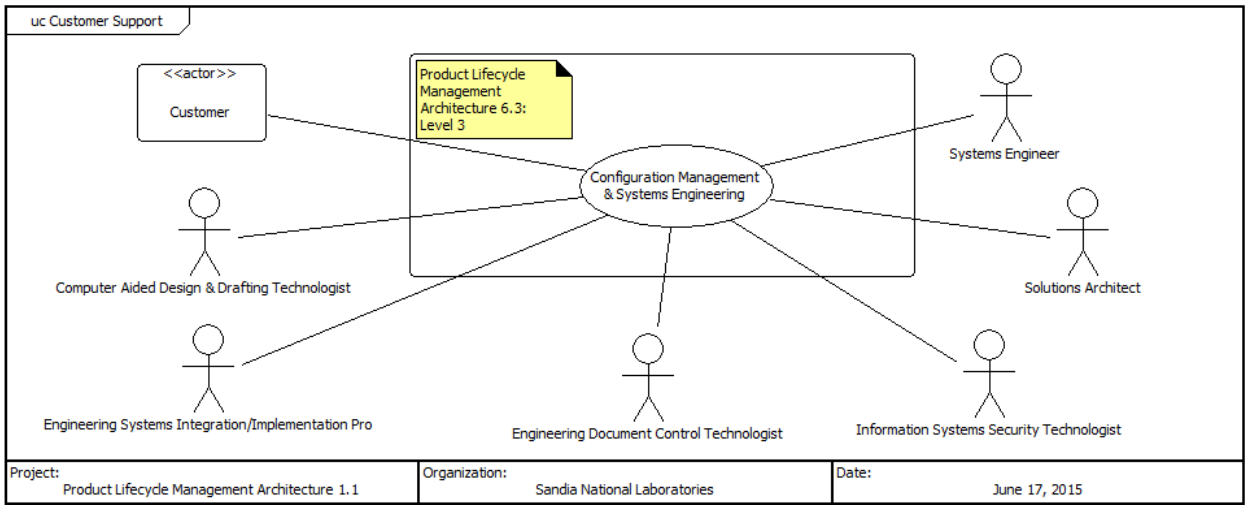


Figure 43. Use Case 3.11: Product Lifecycle Management & Nuclear Weapon 6.3 Level 3

- Level 3 use case 3.12 for NW 6.3: Business Operations. This use case describes the actors involved in providing a known business operations service such as access control to PDMLink and Creo during the 6.3 phase.

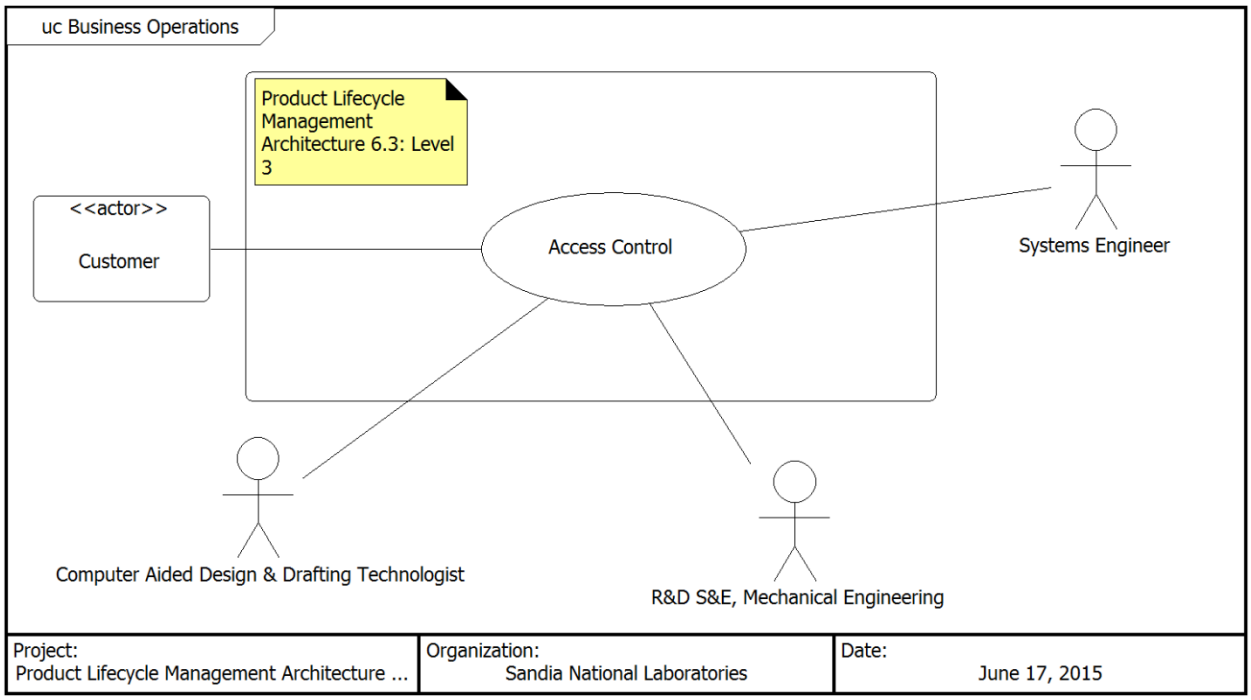


Figure 44. Use Case 3.12: Product Lifecycle Management & Nuclear Weapon 6.3 Level 3

Based on the use cases for PLM and phase 6.3, a sequence diagram was created to show relation to high-level requirements developed for the architecture.

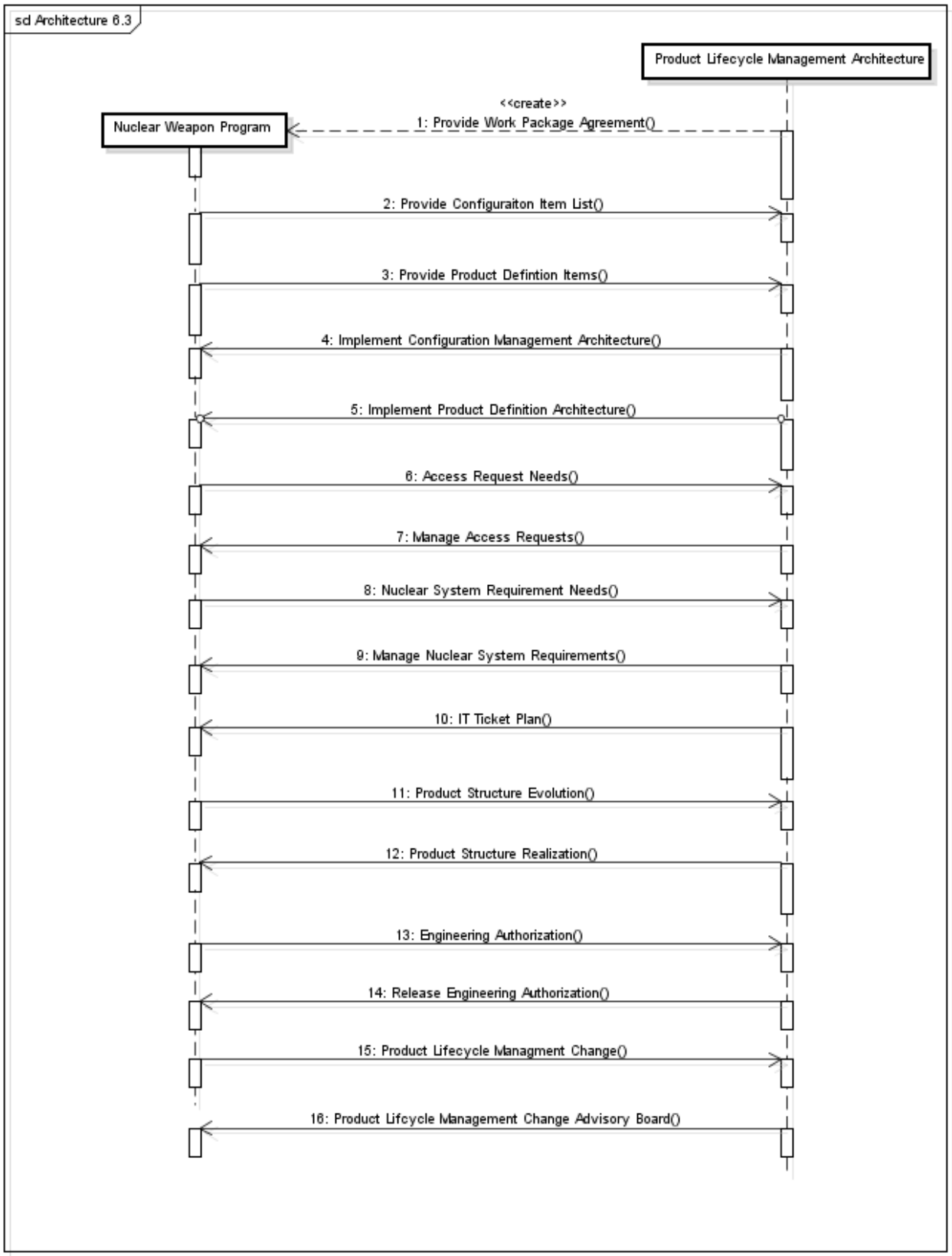


Figure 45. Product Lifecycle Management Architecture Sequence Diagram for Phase 6.3

Based on the above methodology, an execution set of requirements were developed and are shown in Figure 46. The abbreviation for this set is EX.

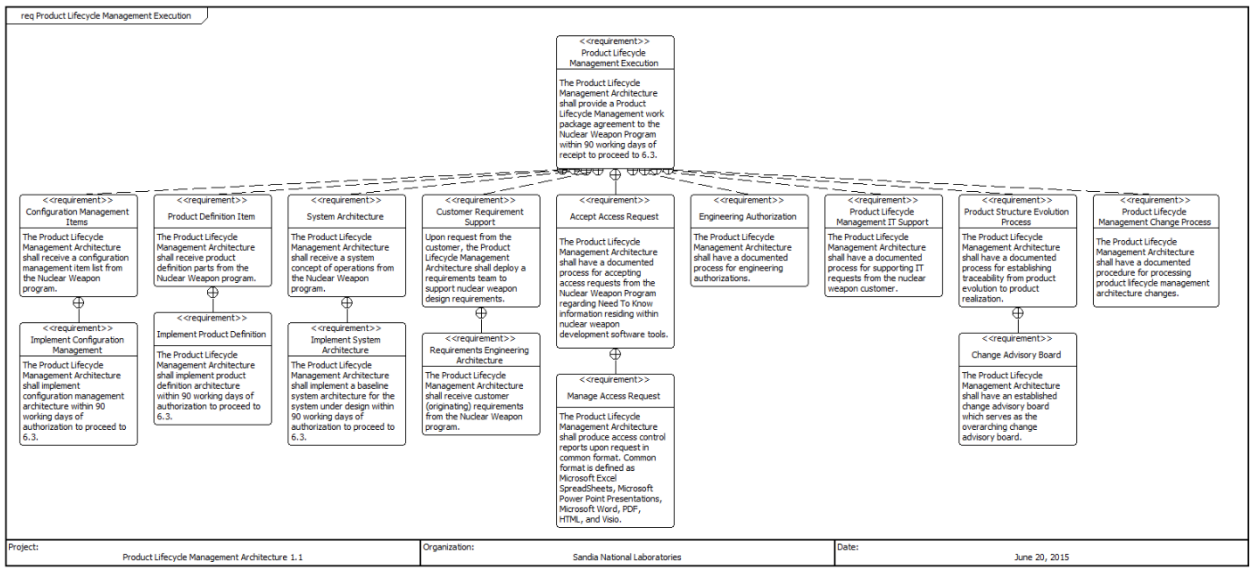


Figure 46. Product Lifecycle Management Execution Set of Requirements

6.3 Product Lifecycle Management Architecture and Nuclear Weapon Phase 6.4

This use-case describes the PLM Architecture in relation to phase 6.4.

- Level 1 use case for NW 6.4: Product Lifecycle Management Architecture Monitoring and Controlling. This series of use cases describes how the PLM will interface with the customer during production engineering.

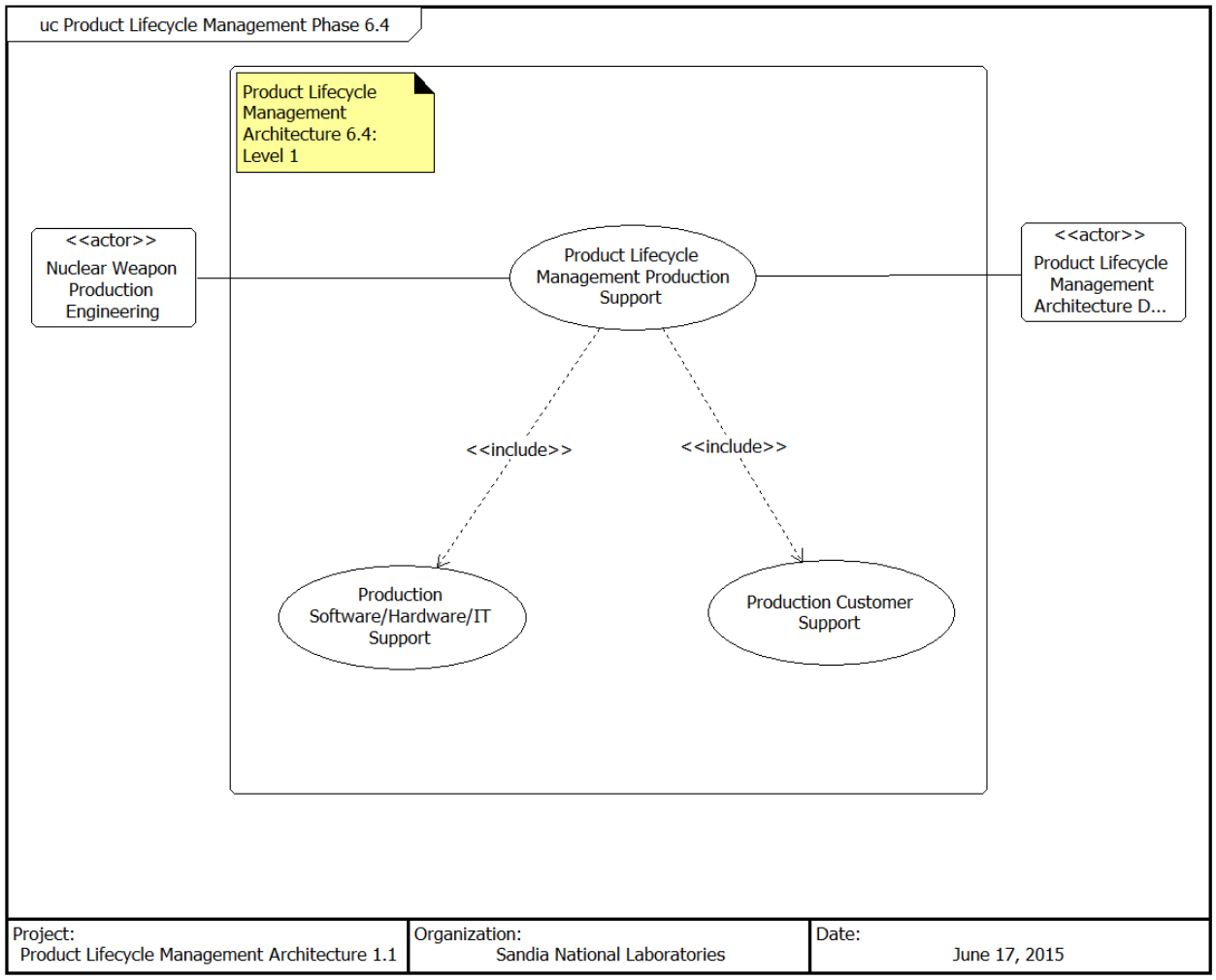


Figure 47. Use Case: Product Lifecycle Management & Nuclear Weapon 6.4 Level 1

Level 2 contains two use cases for NW phase 6.4.

- Level 2 use case 2.1 for NW 6.4: Production Software/Hardware/IT Support. This series of use cases describes how the PLM will interface with the customer during production engineering.

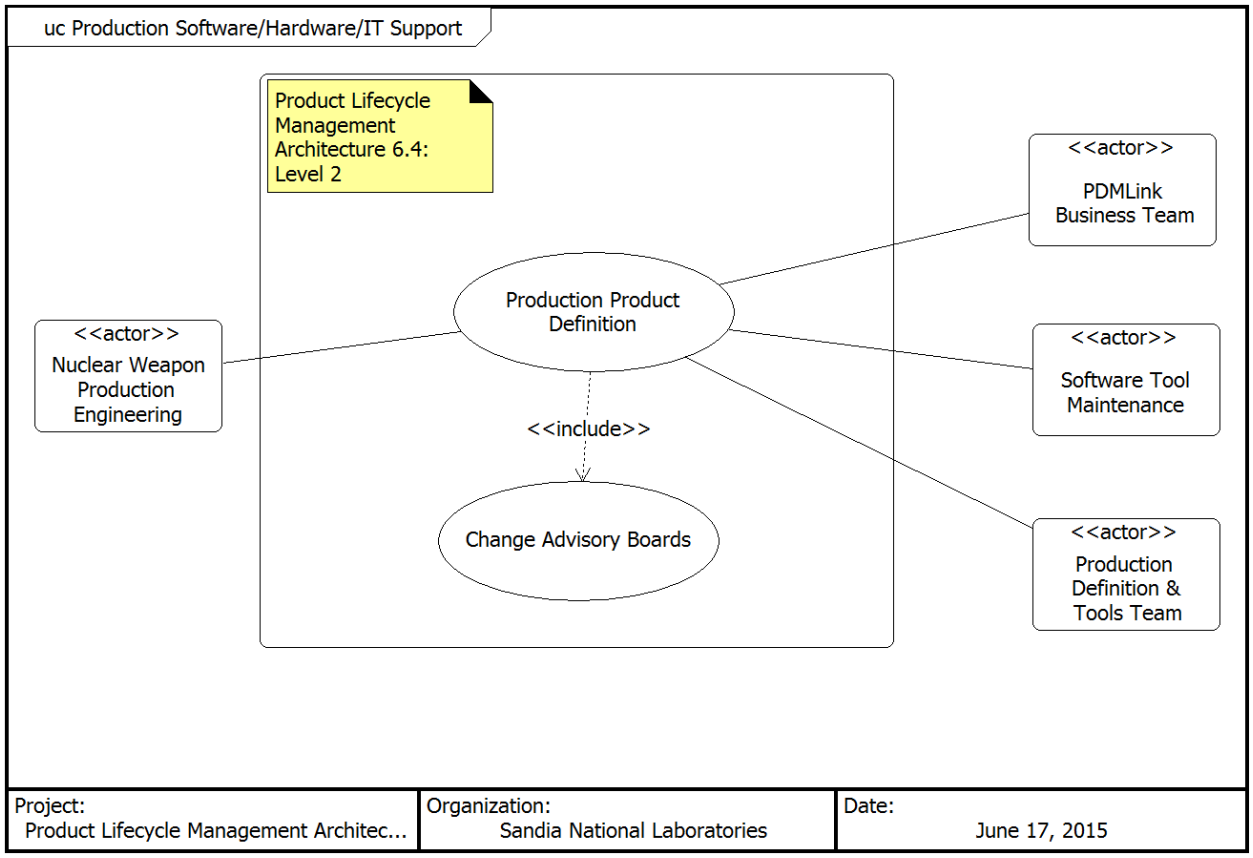


Figure 48. Use Case 2.1: Product Lifecycle Management & Nuclear Weapon 6.4 Level 2

- Level 2 use case 2.2 for NW 6.4: Production Customer Support. This series of use cases describes how the PLM will interface with the customer during production engineering.

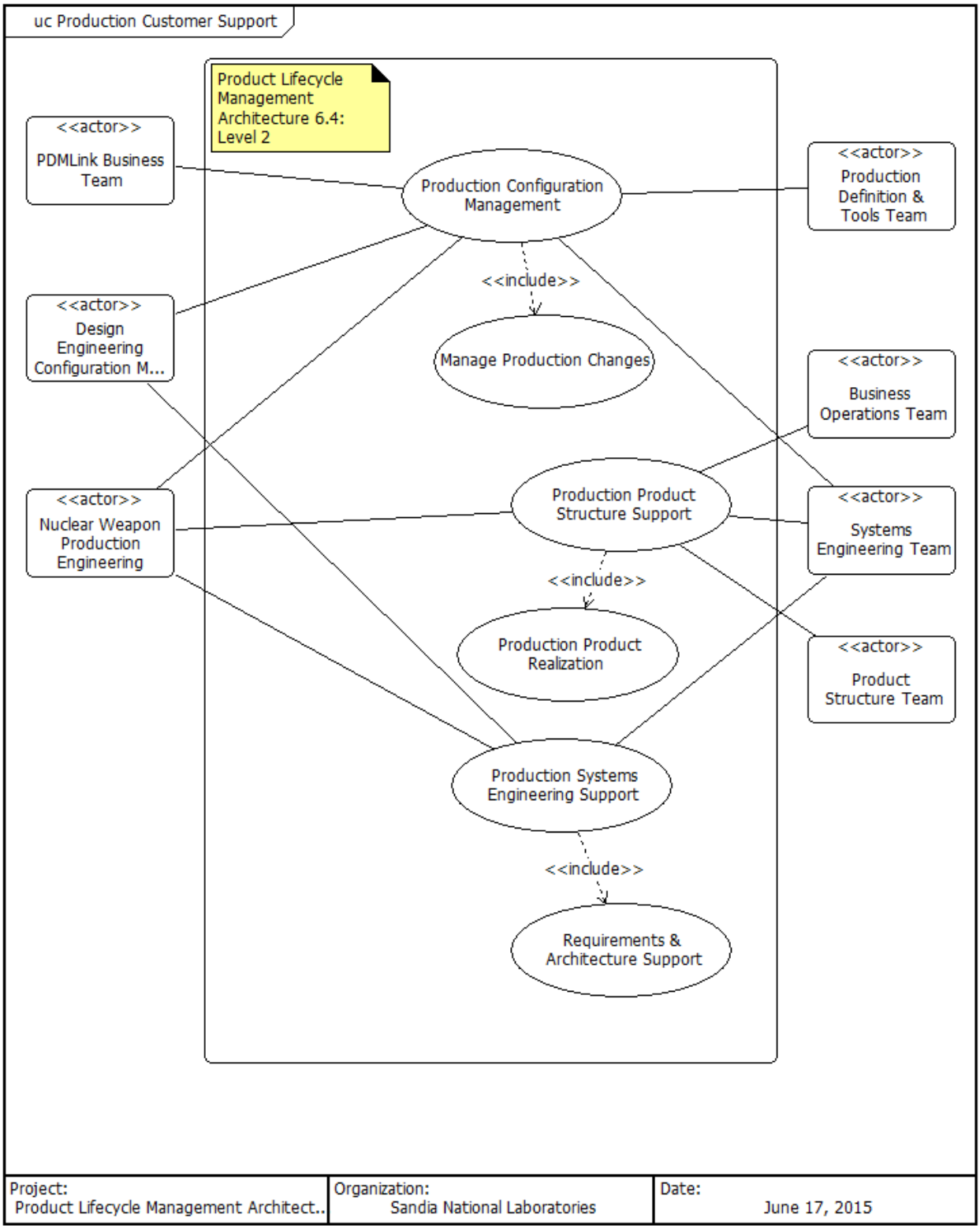


Figure 49. Use Case 2.2: Product Lifecycle Management & Nuclear Weapon 6.4 Level 2

Level 3 contains four use cases for NW phase 6.4.

- Level 3 use case 3.1 for NW 6.4: Change Advisory Boards. This series of use cases describes how the PLM will interface with the customer during production engineering.

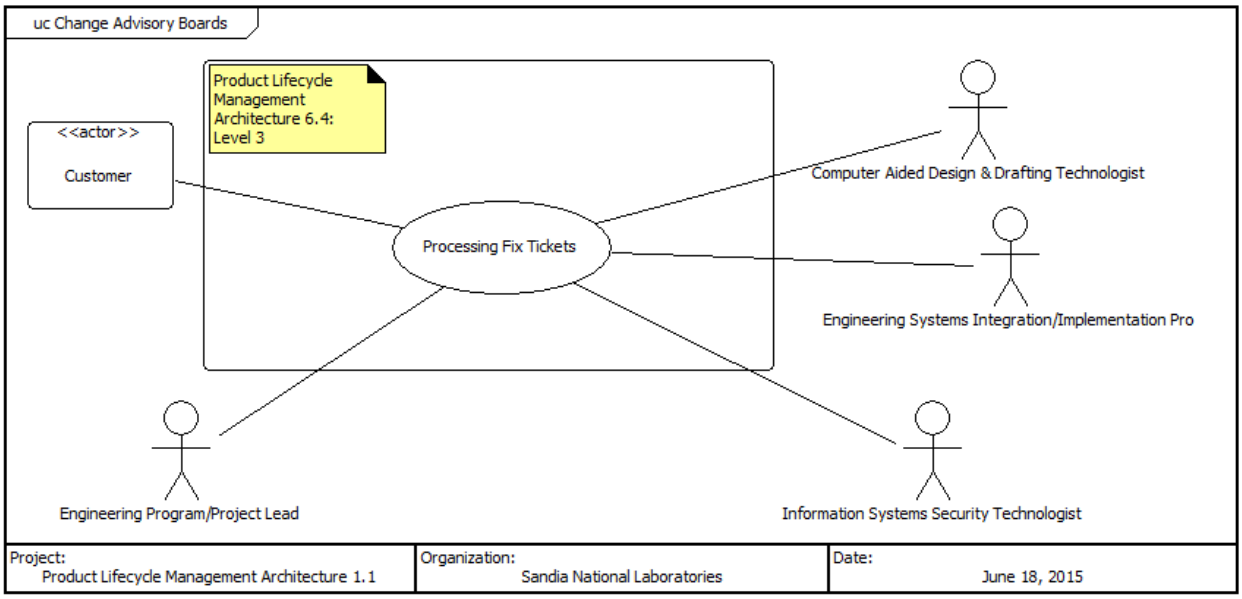


Figure 50. Use Case 3.1: Product Lifecycle Management & Nuclear Weapon 6.4 Level 3

- Level 3 use case 3.2 for NW 6.4: Manage Production Changes. This series of use cases describes how the PLM will interface with the customer during production engineering.

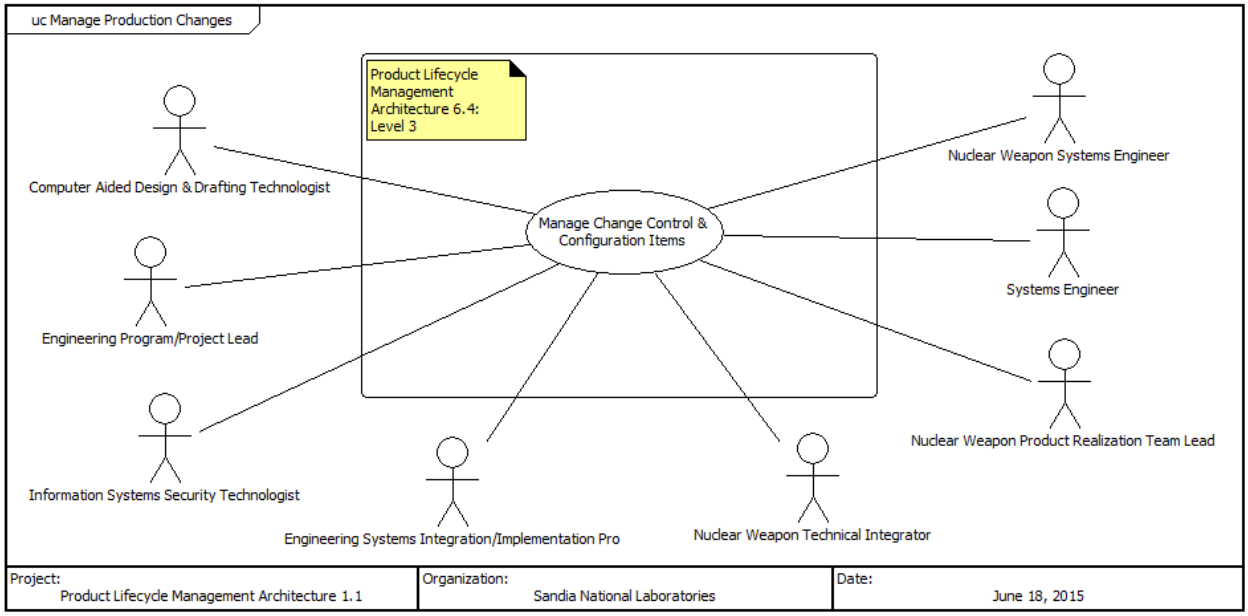


Figure 51. Use Case 3.2: Product Lifecycle Management & Nuclear Weapon 6.4 Level 3

- Level 3 use case 3.3 for NW 6.4: Production Product Realization. This series of use cases describes how the PLM will interface with the customer during production engineering.

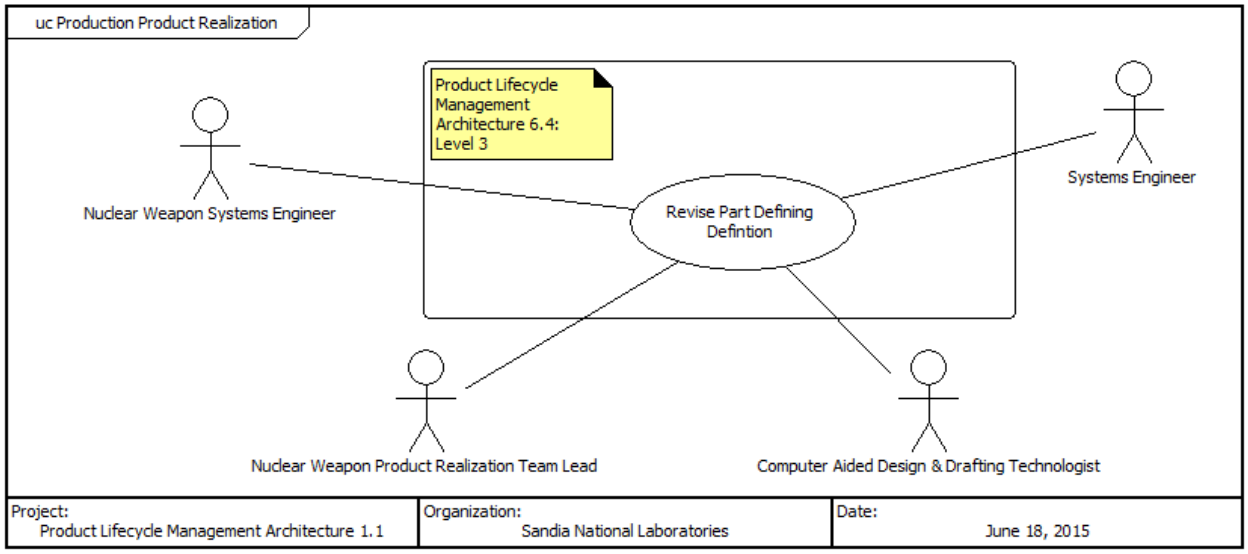


Figure 52. Use Case 3.3: Product Lifecycle Management & Nuclear Weapon 6.4 Level 3

- Level 3 use case 3.4 for NW 6.4: Requirements & Architecture Support. This series of use cases describes how the PLM will interface with the customer during production engineering.

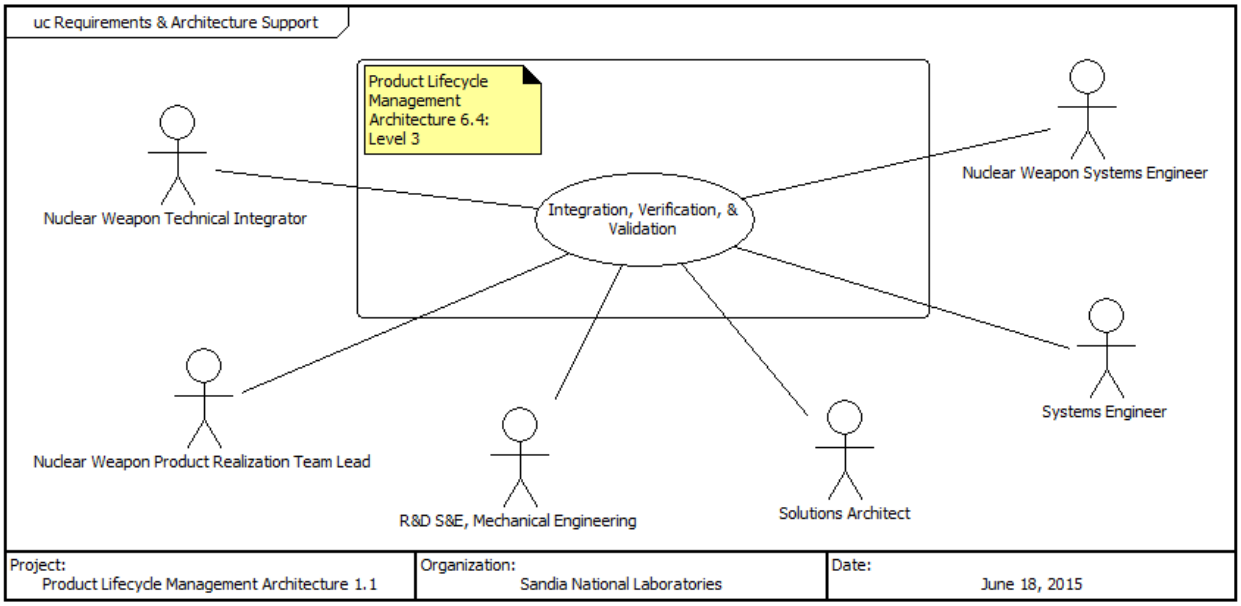


Figure 53. Use Case 3.4: Product Lifecycle Management & Nuclear Weapon 6.4 Level 3

Based on the use-cases for PLM and phase 6.4, a sequence diagram was created to show relation to high-level requirements developed for the architecture.

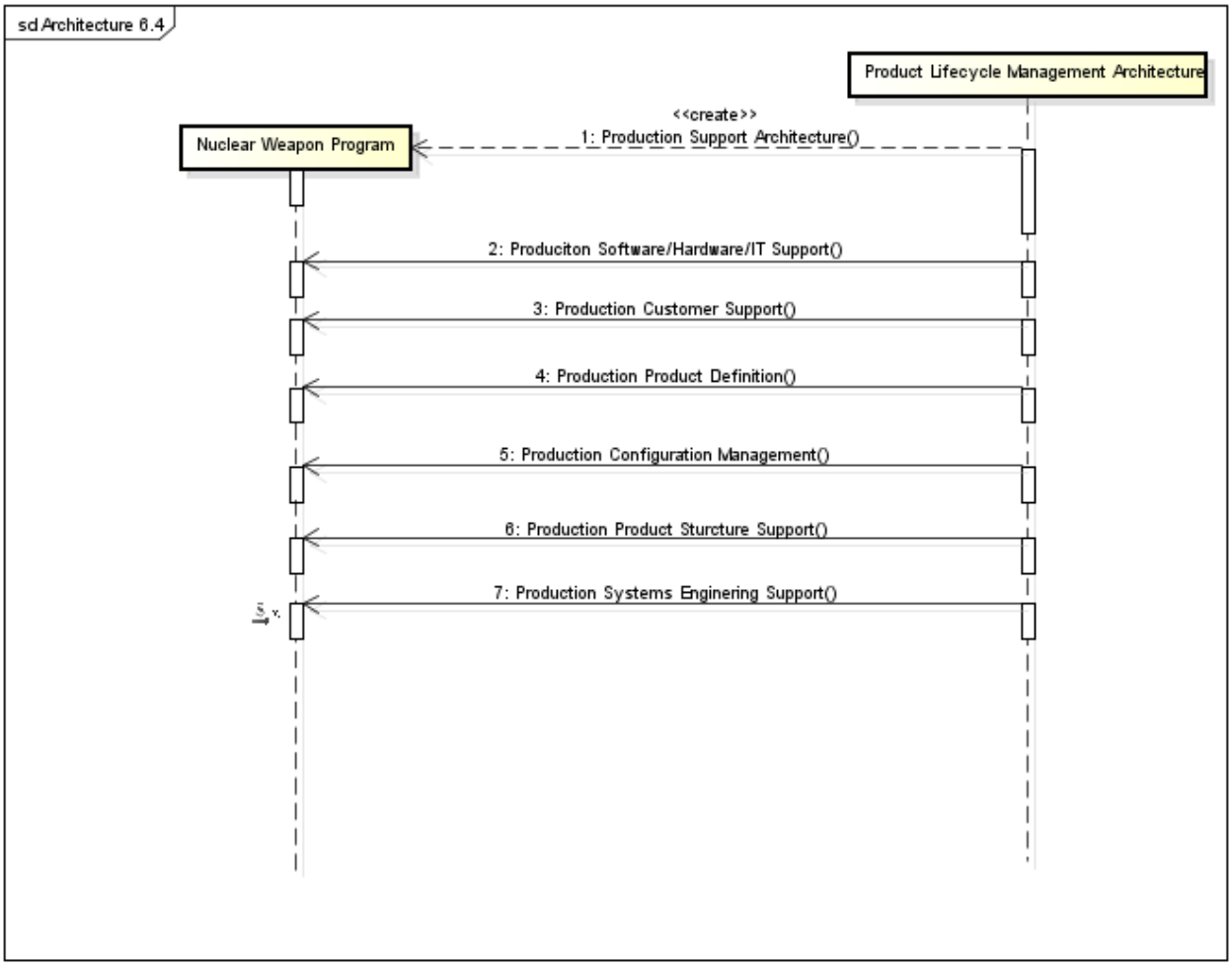


Figure 54. Product Lifecycle Management Architecture Sequence Diagram for Phase 6.4

Based on the above methodology, a monitoring & controlling set of requirements were developed and are shown in Figure 55. The abbreviation for this set is MC.

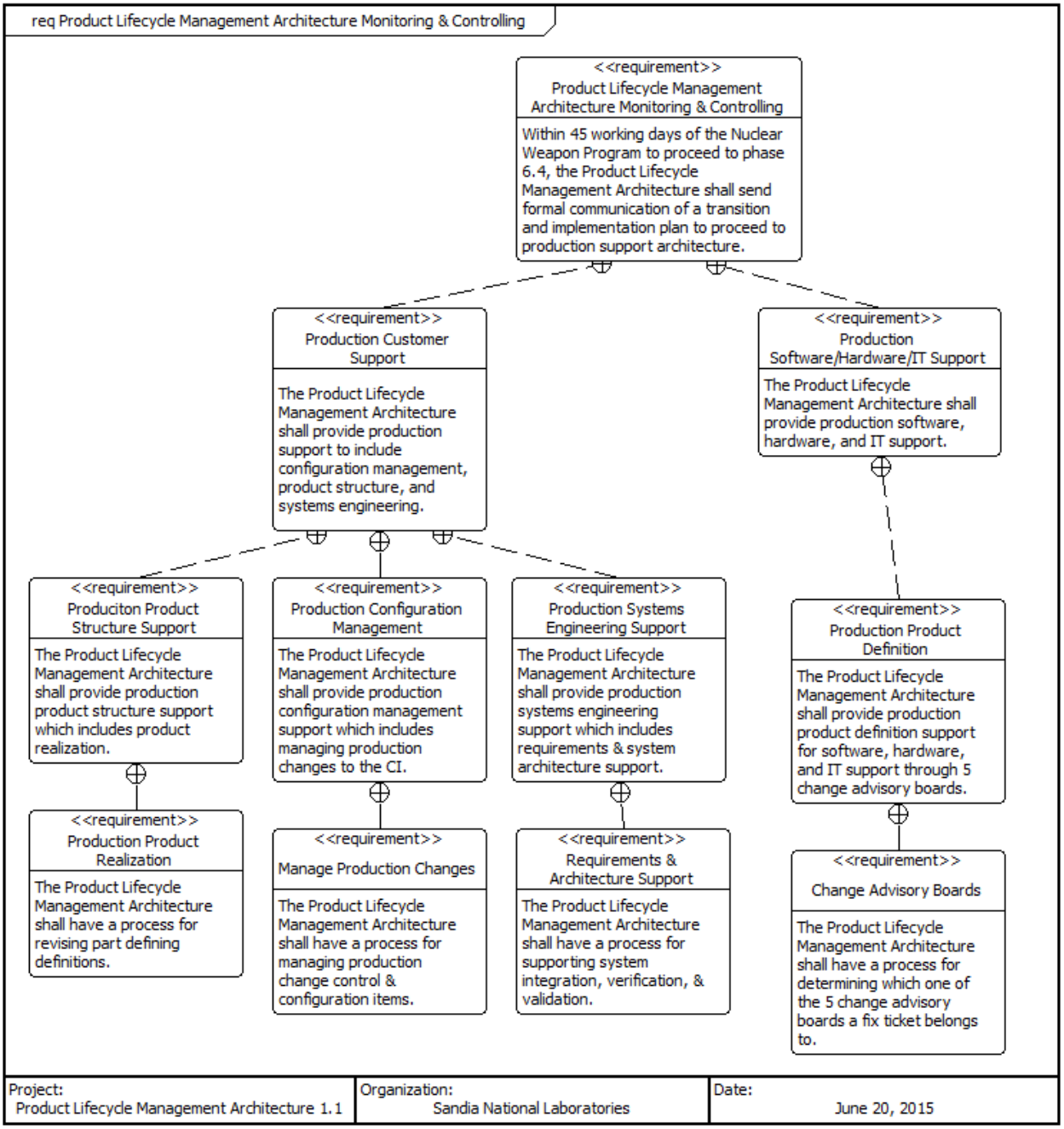


Figure 55. Product Lifecycle Management Monitoring & Controlling Set of Requirements

6.4 Product Lifecycle Management Architecture and Nuclear Weapon Phase 6.5

This use-case describes the PLM Architecture in relation to phase 6.5.

- Level 1 use case for NW 6.5: Quality Control. This series of use cases describes how the PLM will interface with the customer during First Production Unit (FPU) by providing quality control support.

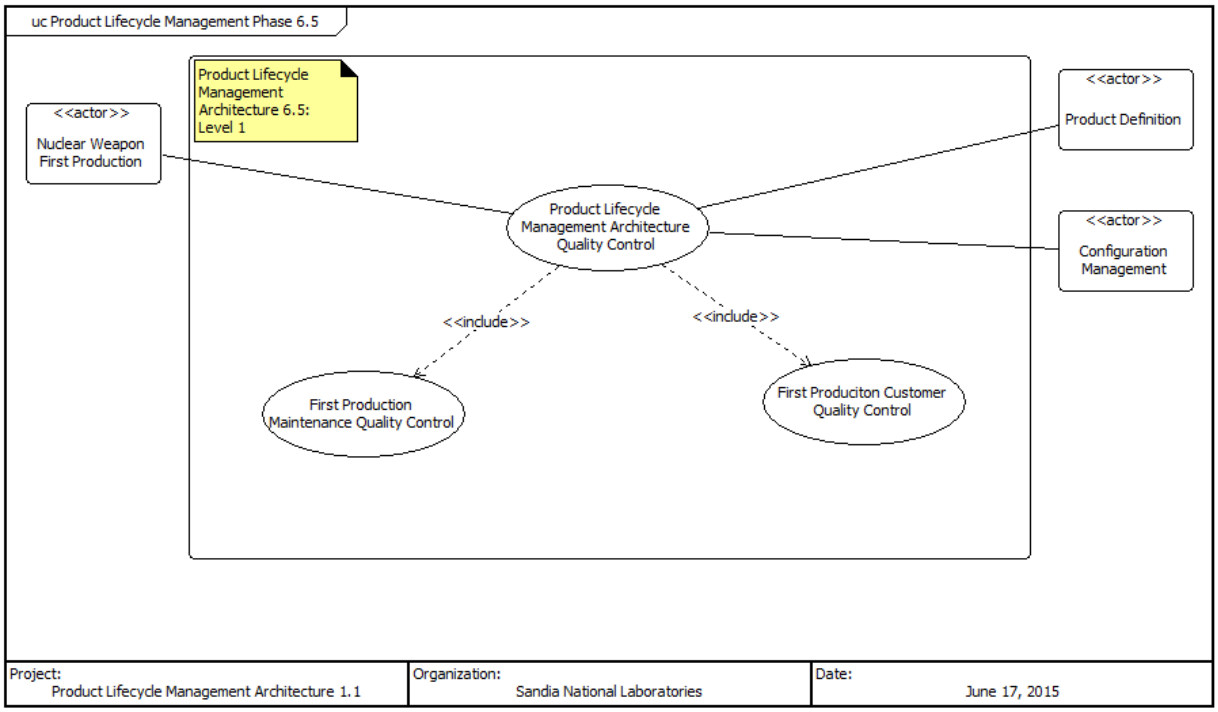


Figure 56. Use Case: Product Lifecycle Management & Nuclear Weapon 6.5 Level 1

Level 2 contains two use cases for NW phase 6.5.

- Level 2 use case 2.1 for NW 6.5: First Production Maintenance Quality Control. This series of use cases describes how the PLM will interface with the customer during FPU.

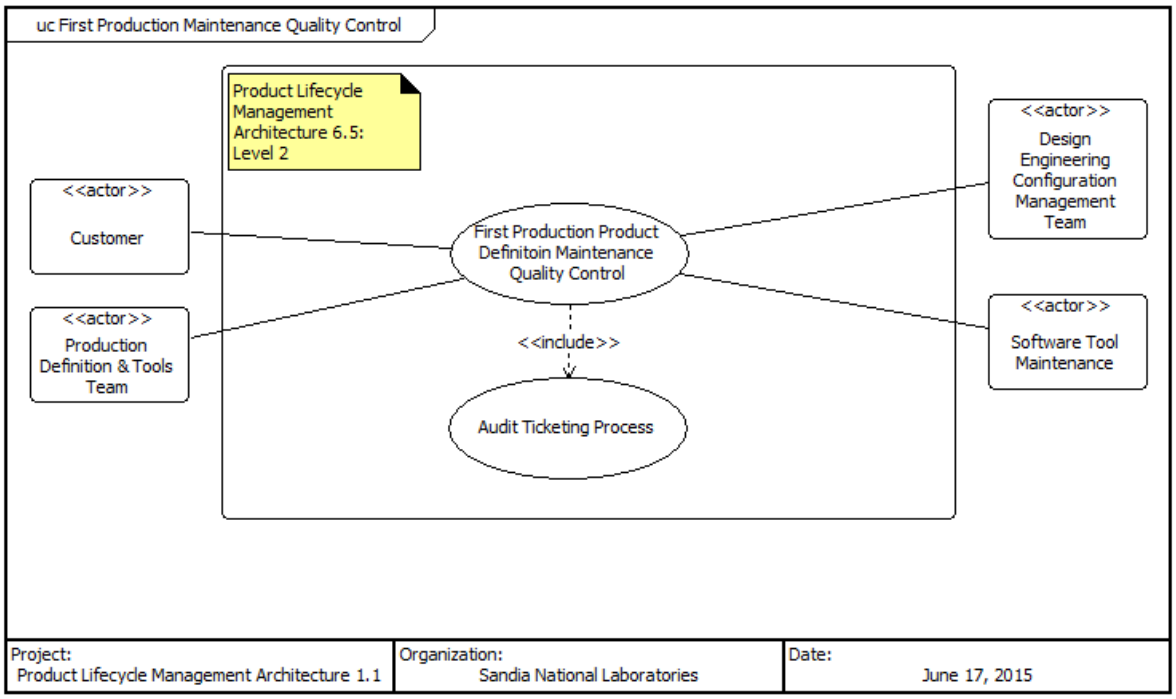


Figure 57. Use Case 2.1: Product Lifecycle Management & Nuclear Weapon 6.5 Level 2

- Level 2 use case 2.2 for NW 6.5: First Production Customer Quality Control. This series of use cases describes how the PLM will interface with the customer during FPU.

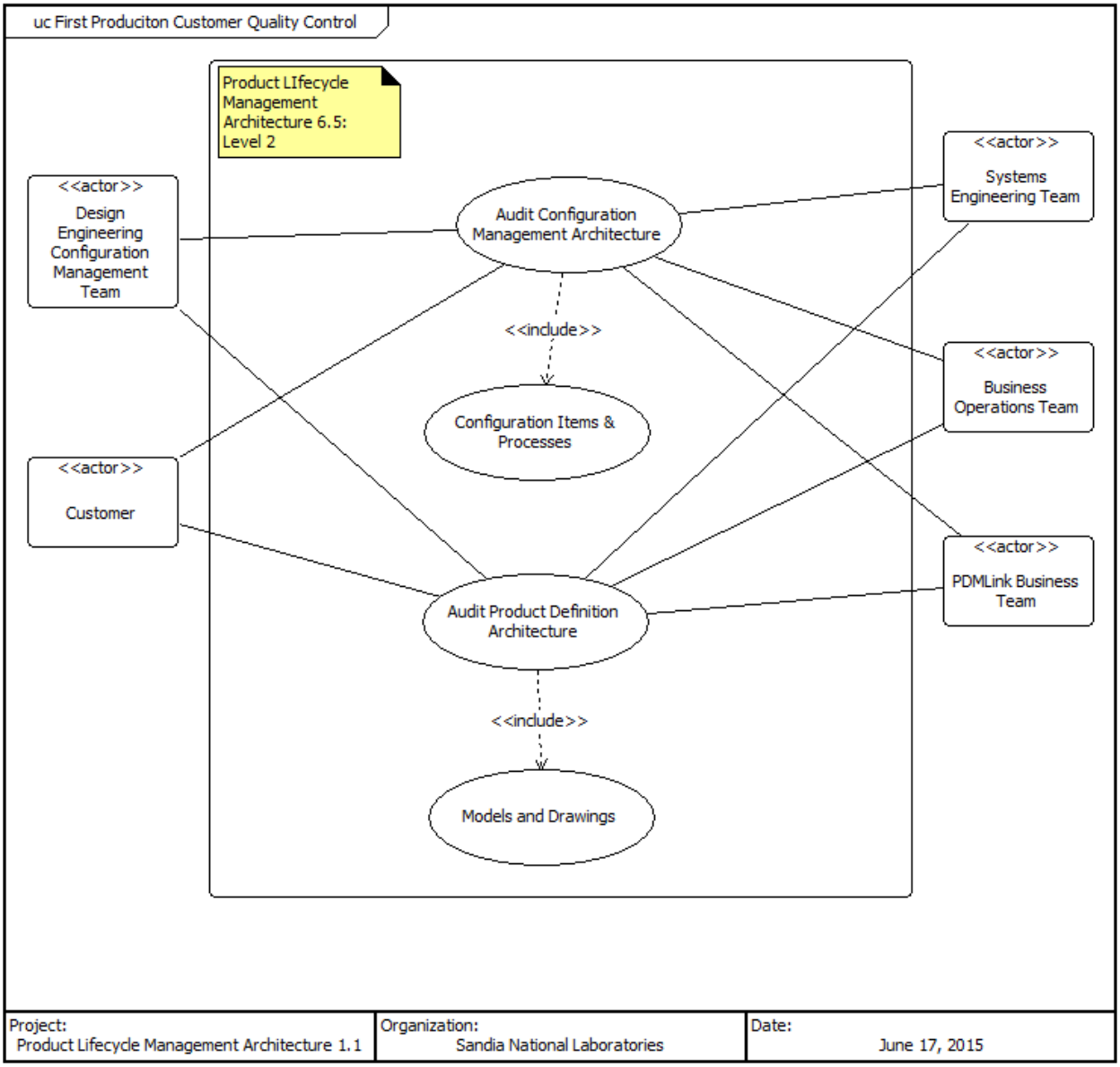


Figure 58. Use Case 2.2: Product Lifecycle Management & Nuclear Weapon 6.5 Level 2

Level 3 contains three use cases for NW phase 6.5.

- Level 3 use case 3.1 for NW 6.5: Audit Ticking Process. This series of use cases describes how the PLM will interface with the customer during FPU.

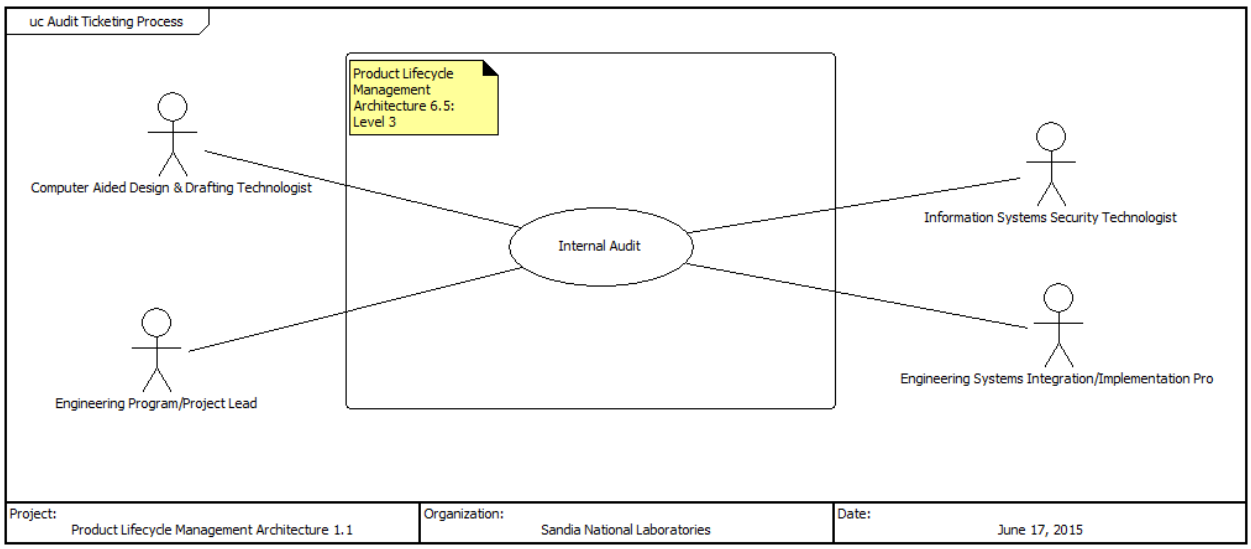


Figure 59. Use Case 3.1: Product Lifecycle Management & Nuclear Weapon 6.5 Level 3

- Level 3 use case 3.2 for NW 6.5: Configuration Items & Processes. This series of use cases describes how the PLM will interface with the customer during FPU.

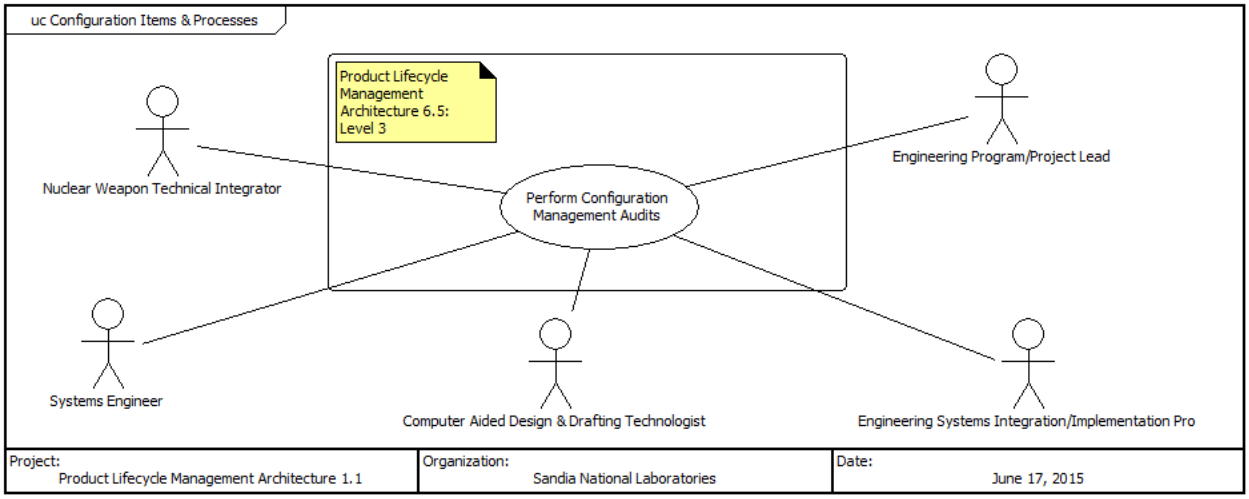


Figure 60. Use Case 3.2: Product Lifecycle Management & Nuclear Weapon 6.5 Level 3

- Level 3 use case 3.3 for NW 6.5: Models and Drawings. This series of use cases describes how the PLM will interface with the customer during FPU.

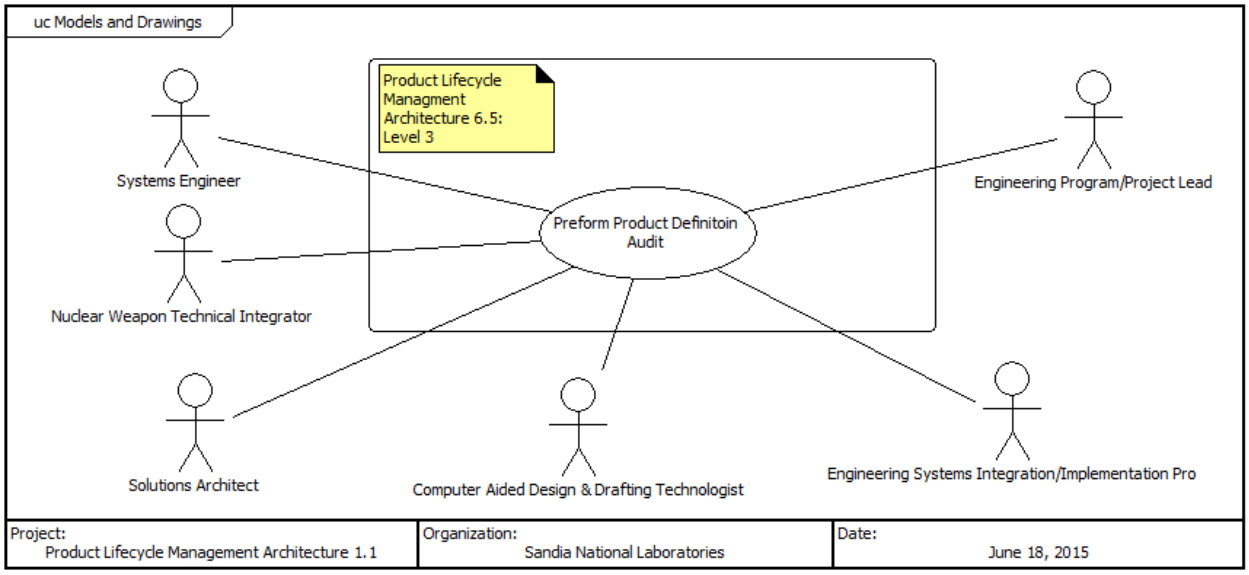


Figure 61. Use Case 3.3: Product Lifecycle Management & Nuclear Weapon 6.5 Level 3

Based on the use cases for PLM and phase 6.5, a sequence diagram was created to show relation to high-level requirements developed for the architecture.

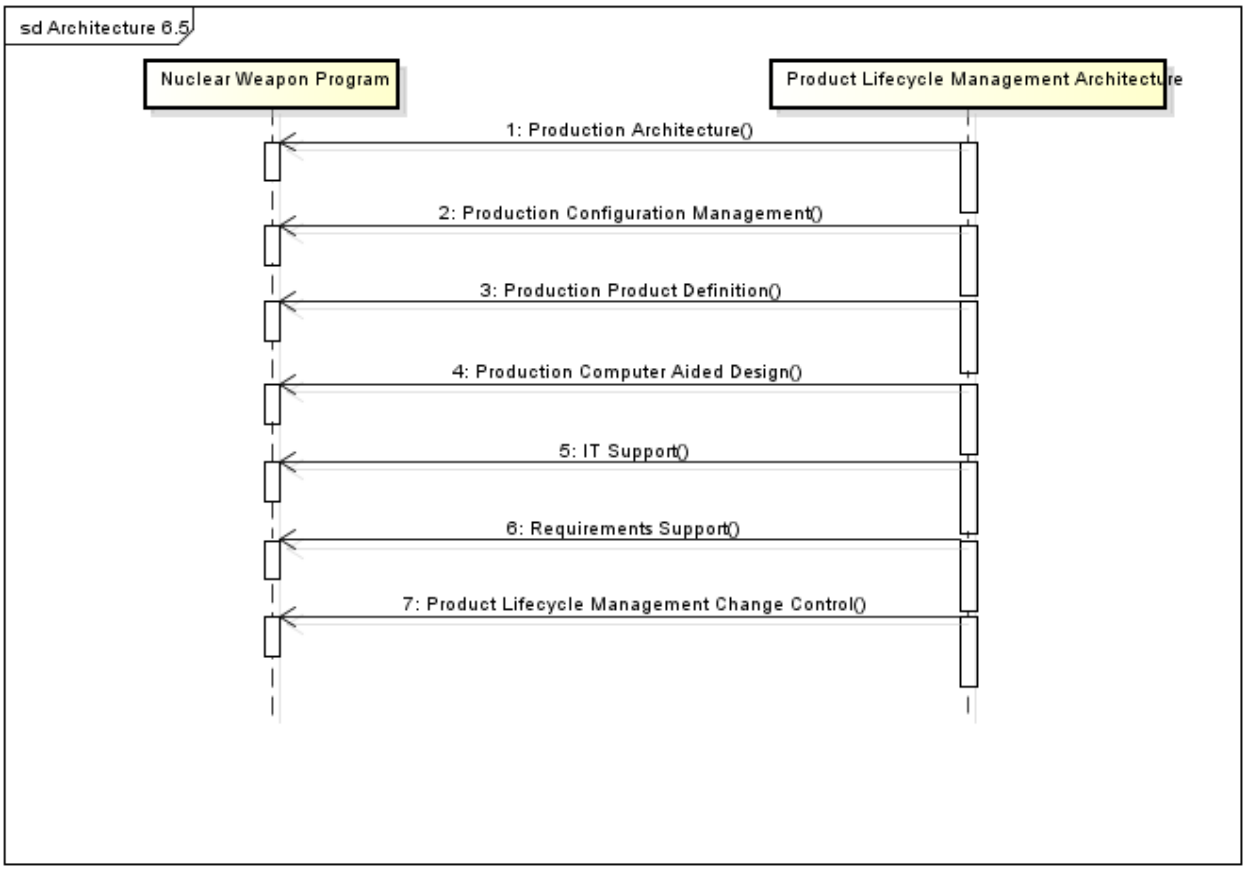


Figure 62. Product Lifecycle Management Architecture Sequence Diagram for Phase 6.5

Based on the above methodology, a quality control set of requirements were developed and are shown in Figure 63. The abbreviation for this set is QC.

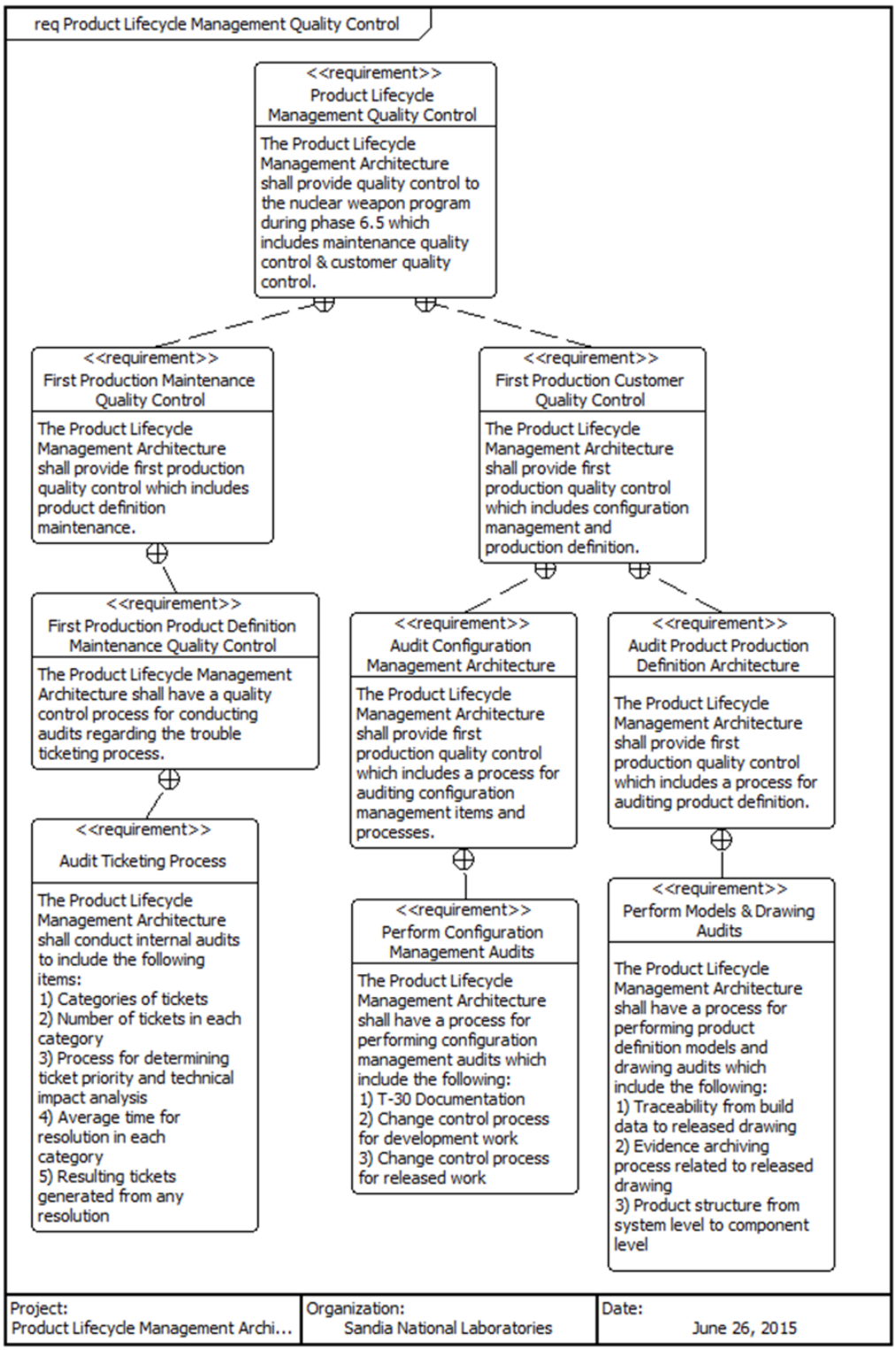


Figure 63. Product Lifecycle Management Quality Control Set of Requirements

6.5 Product Lifecycle Management Architecture and Nuclear Weapon Phase 6.6-7

This use case describes the PLM Architecture in relation to phase 6.6-7.

Level 1 use case for NW 6.6-7: Closing. This series of use cases describes how the PLM will interface with the customer during quality production and dismantlement by providing end of lifecycle support.

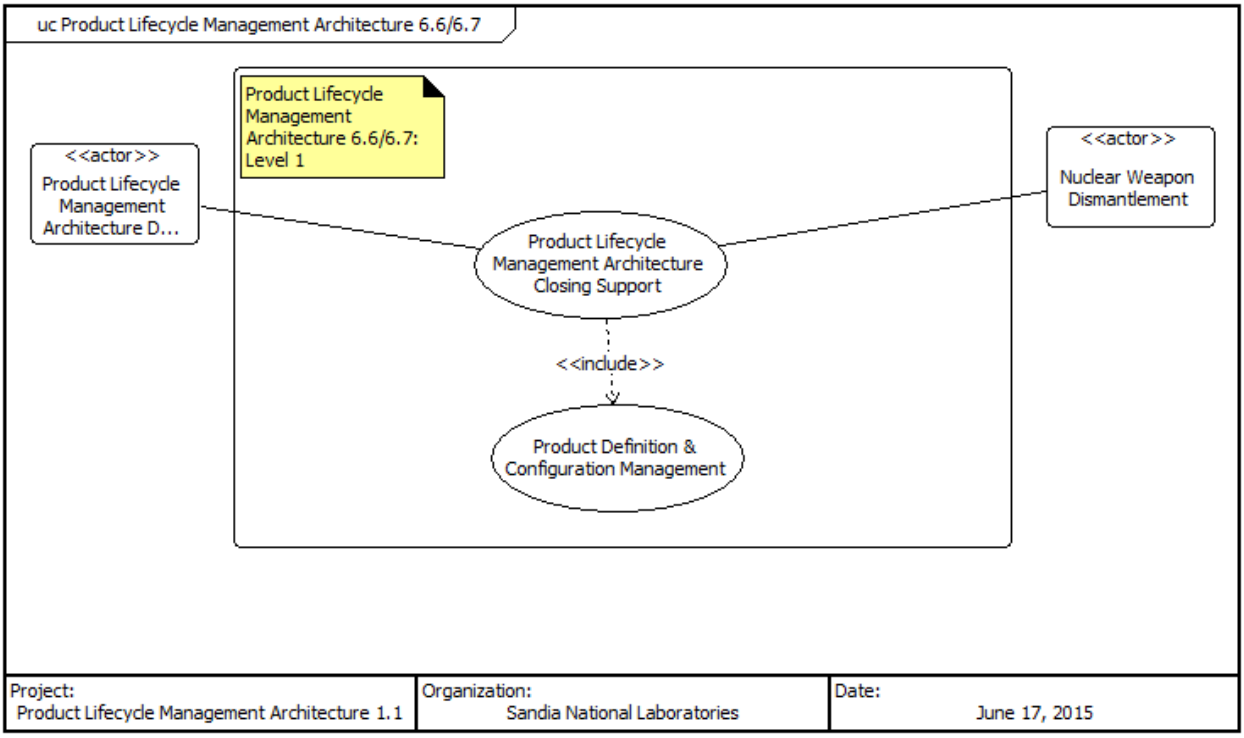


Figure 64. Use Case: Product Lifecycle Management & Nuclear Weapon 6.6-7 Level 1

- Level 2 use case for NW 6.6-7: Product Definition & Configuration Management. This series of use cases describes how the PLM will interface with the customer during product end of lifecycle.

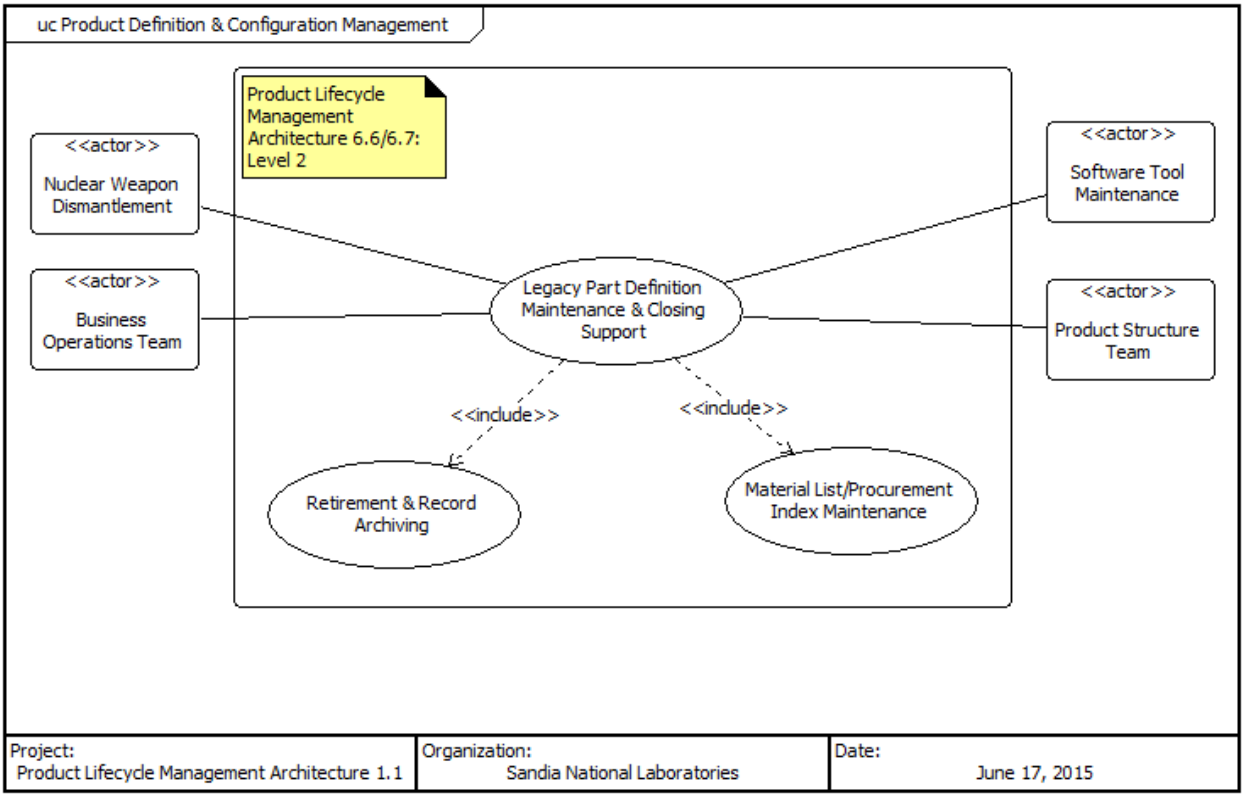


Figure 65. Use Case: Product Lifecycle Management & Nuclear Weapon 6.6-7 Level 2

Level 3 contains two use cases for NW phase 6.6-7.

- Level 3 use case 3.1 for NW 6.6-7: Retirement & Record Archiving. This series of use cases describes how the PLM will interface with the customer during product end of lifecycle.

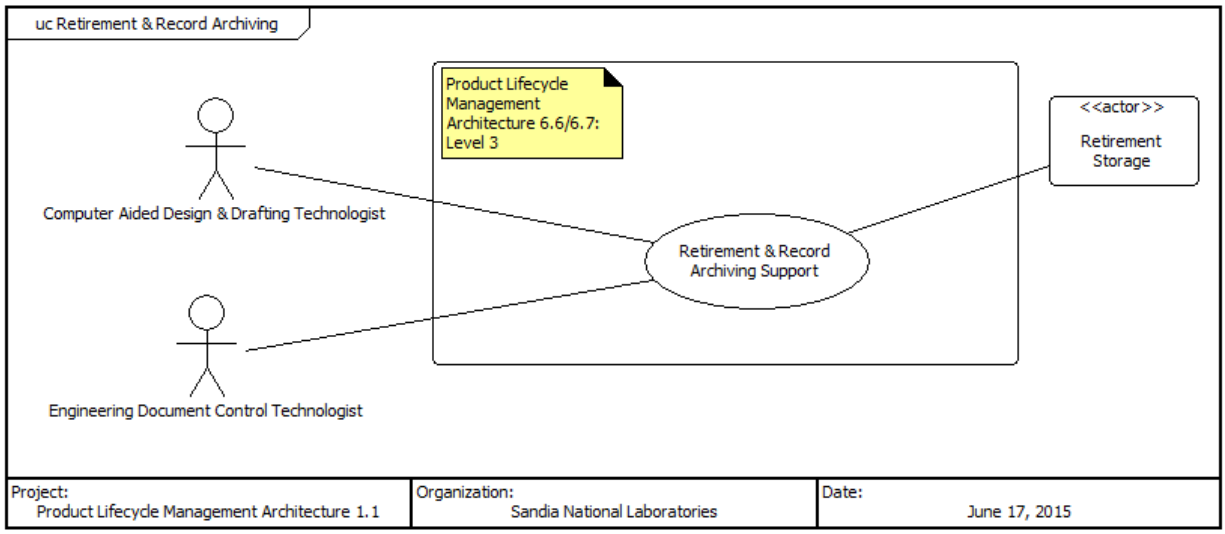


Figure 66. Use Case 3.1: Product Lifecycle Management & Nuclear Weapon 6.6-7 Level 3

- Level 3 use case 3.2 for NW 6.6-7: Material List/Procurement Index Maintenance. This series of use cases describes how the PLM will interface with the customer during product end of lifecycle.

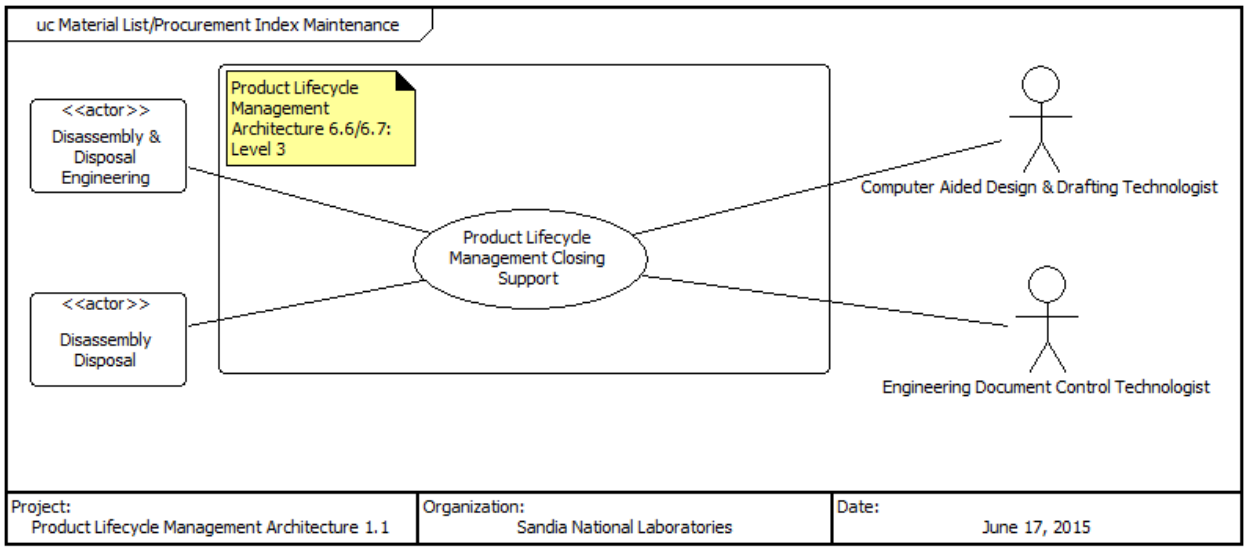


Figure 67. Use-Case 3.2: Product Lifecycle Management & Nuclear Weapon 6.6-7 Level 3

Based on the use cases for PLM and Phase 6.6-7, a sequence diagram was created to show relation to high-level requirements developed for the architecture.

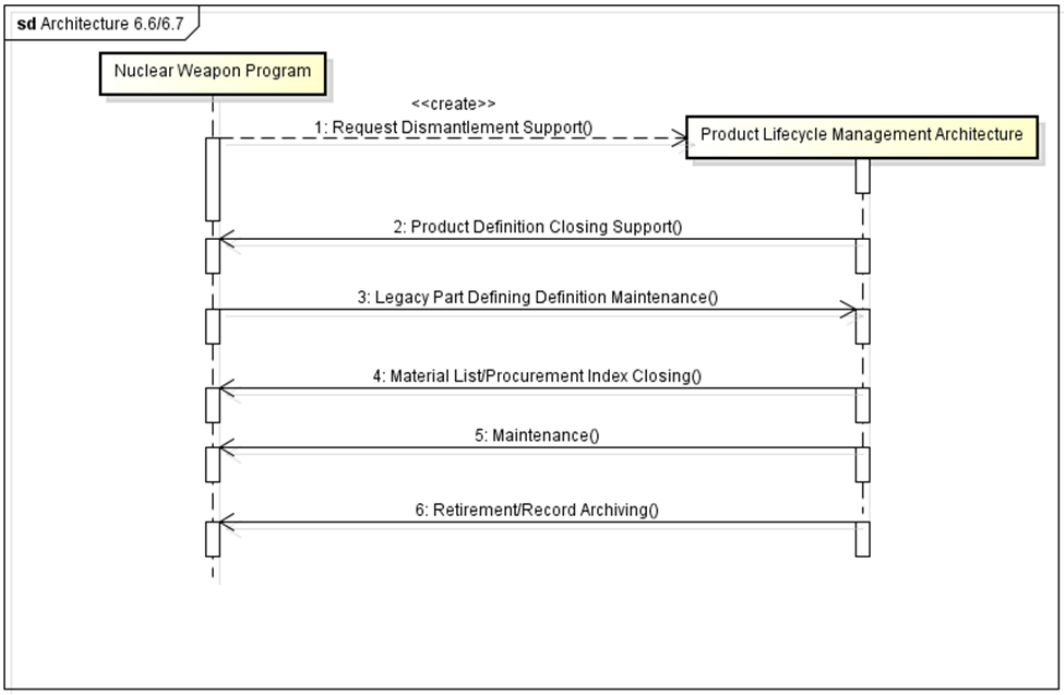


Figure 68. Product Lifecycle Management Architecture Sequence Diagram for Phase 6.6-7

Based on the above methodology, a closing set of requirements were developed and are shown in Figure 69. The abbreviation for this set is CL.

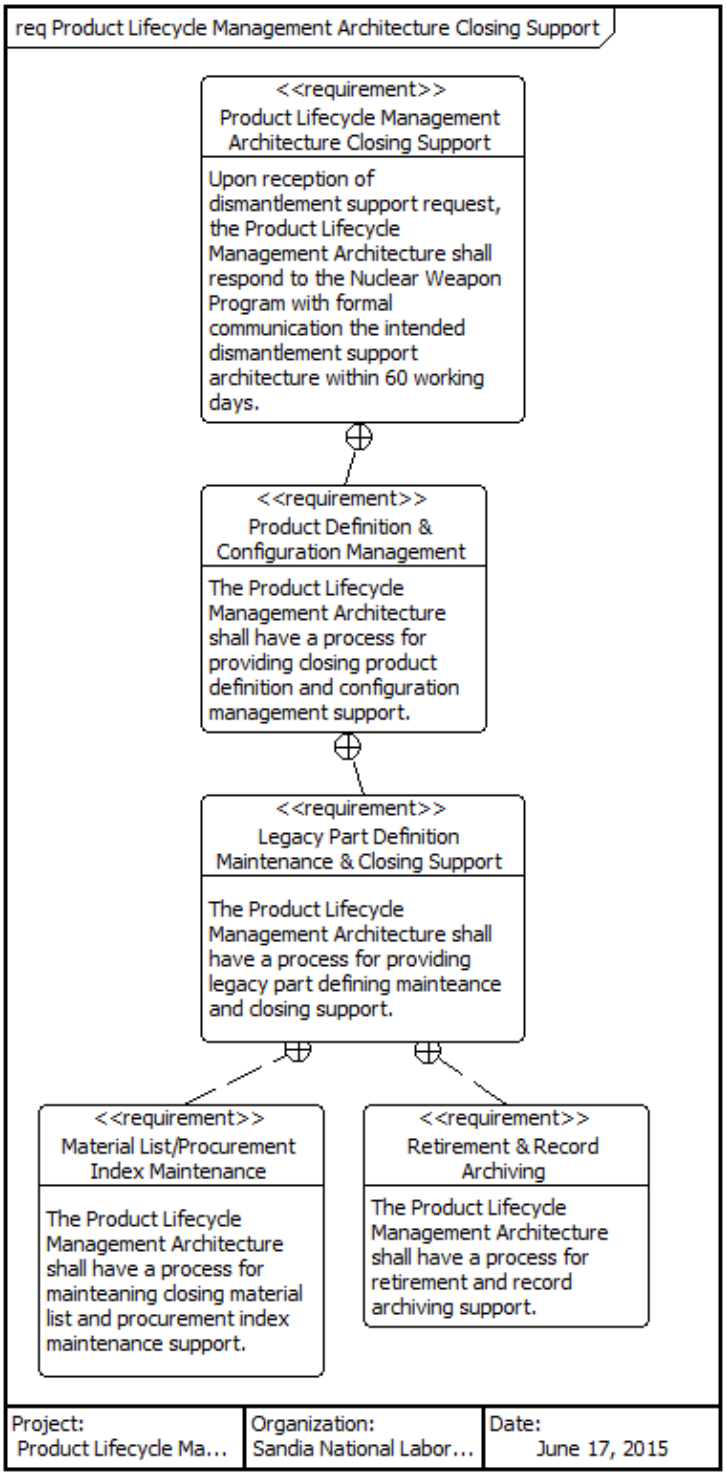


Figure 69. Product Lifecycle Management Closing Set of Requirements

7 SOLUTION SPACE: PRODUCT LIFECYCLE MANAGEMENT ALLOCATED ARCHITECTURE DEVELOPMENT

This section describes the logical architecture, the allocation of requirements to the logical architecture, the physical architecture, and the allocation of the logical architecture to the physical architecture. In addition to each level of logical architecture, interfaces were developed and are captured in internal block diagram views. Figure 70 shows the expanded view of the three levels of the logical architecture. This section will provide details for each level of the logical architecture.

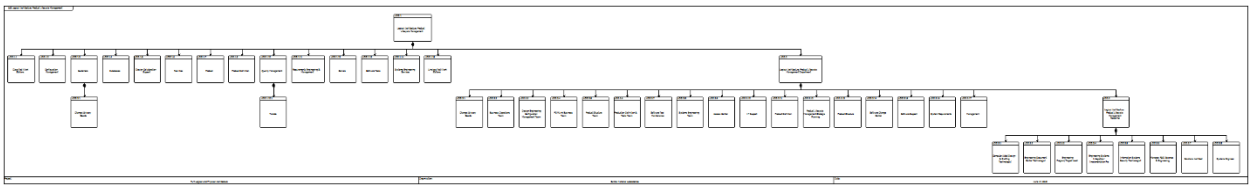


Figure 70. Block Definition of Logical Architecture Expanded View

7.1 Logical Architecture for Product Lifecycle Management Level 1

LOG.1 Logical Architecture: Product Lifecycle Management Level 1

Type: System Architecture

Built From Lower-Level Component(s):

- LOG.1.1 Classified Work Stations
- LOG.1.2 Configuration Management
- LOG.1.3 Customers
- LOG.1.4 Databases
- LOG.1.5 Design Collaboration Support
- LOG.1.6 Facilities
- LOG.1.7 Product
- LOG.1.8 Product Definition
- LOG.1.10 Quality Management
- LOG.1.11 Requirements Engineering & Management
- LOG.1.12 Servers
- LOG.1.13 Software Tools
- LOG.1.14 Systems Engineering Services
- LOG.1.15 Unclassified Work Stations
- LOG.2 Logical Architecture: Product Lifecycle Management Department

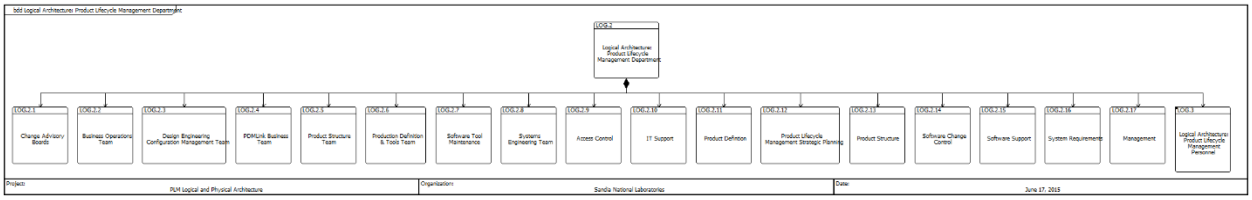


Figure 71. Block Definition of Logical Architecture Level 1

7.1.1 Internal Block Diagram for Logical Architecture Level 1

The yellow box in Figure 72 shows the interfaces of the four main services provided by the PLM.

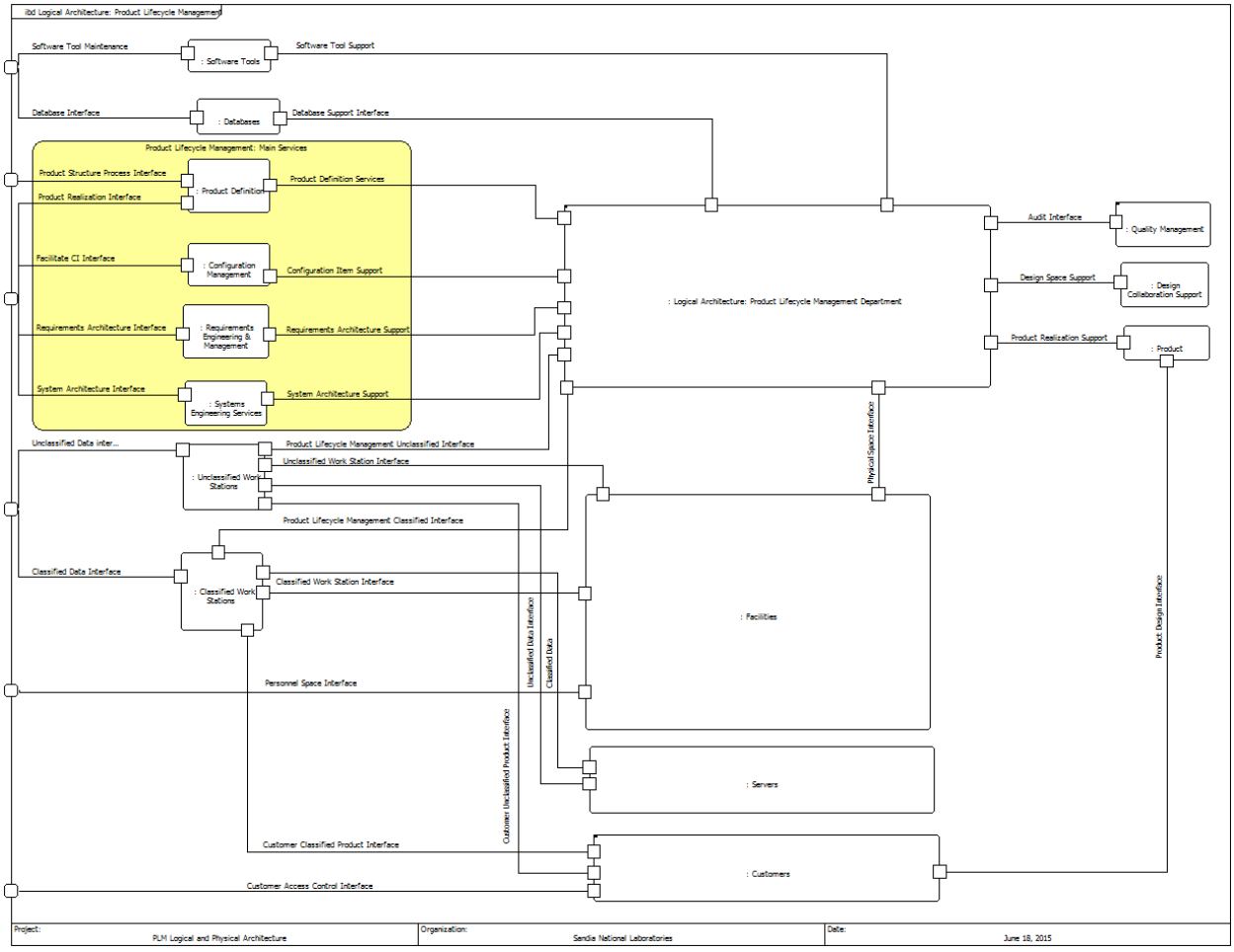


Figure 72. Internal Block Diagram for Logical Architecture Level 1

Below are the interfce details for level 1 interfaces.

LOG.1.1 Classified Work Stations

Built In Higher-Level Component(s):

LOG.1 Logical Architecture: Product Lifecycle Management

Joined To Logical Interface:

Classified Data
Classified Data Interface
Classified Work Station Interface
Customer Classified Product Interface
Product Lifecycle Management Classified Interface

LOG.1.2 Configuration Management

Built In Higher-Level Component(s):
LOG.1 Logical Architecture: Product Lifecycle Management

Joined To Logical Interface:
Configuration Item Support
Facilitate CI Interface

LOG.1.3 Customers

Built In Higher-Level Component(s):
LOG.1 Logical Architecture: Product Lifecycle Management

Built From Lower-Level Component(s):
LOG.2.1 Change Advisory Boards

Joined To Logical Interface:
Customer Access Control Interface
Customer Classified Product Interface
Customer Unclassified Product Interface
Product Design Interface

LOG.1.4 Databases

Built In Higher-Level Component(s):
LOG.1 Logical Architecture: Product Lifecycle Management

Joined To Logical Interface:
Database Interface
Database Support Interface

LOG.1.5 Design Collaboration Support

Built In Higher-Level Component(s):
LOG.1 Logical Architecture: Product Lifecycle Management

Joined To Logical Interface:
Design Space Support

LOG.1.6 Facilities

Built In Higher-Level Component(s):
LOG.1 Logical Architecture: Product Lifecycle Management

Joined To Logical Interface:
Classified Work Station Interface
Personnel Space Interface
Physical Space Interface
Unclassified Work Station Interface

LOG.1.7 Product

Built In Higher-Level Component(s):
LOG.1 Logical Architecture: Product Lifecycle Management

Joined To Logical Interface:

Product Design Interface
Product Realization Support

LOG.1.8 Product Definition

Built In Higher-Level Component(s):
LOG.1 Logical Architecture: Product Lifecycle Management

Joined To Logical Interface:
Product Definition Services
Product Realization Interface
Product Structure Process Interface

LOG.1.10 Quality Management

Built In Higher-Level Component(s):
LOG.1 Logical Architecture: Product Lifecycle Management

Built From Lower-Level Component(s):
LOG.1.10.1 Tickets

Joined To Logical Interface:
Audit Interface

LOG.1.10.1 Tickets

Built In Higher-Level Component(s):
LOG.1.10 Quality Management

Joined To Logical Interface:
Review Trouble Tickets

LOG.1.11 Requirements Engineering & Management

Built In Higher-Level Component(s):
LOG.1 Logical Architecture: Product Lifecycle Management

Joined To Logical Interface:
Requirements Architecture Interface
Requirements Architecture Support

LOG.1.12 Servers

Built In Higher-Level Component(s):
LOG.1 Logical Architecture: Product Lifecycle Management

Joined To Logical Interface:
Classified Data
Unclassified Data Interface

LOG.1.13 Software Tools

Built In Higher-Level Component(s):
LOG.1 Logical Architecture: Product Lifecycle Management

Joined To Logical Interface:
Software Tool Maintenance

Software Tool Support

LOG.1.14 Systems Engineering Services

Built In Higher-Level Component(s):
LOG.1 Logical Architecture: Product Lifecycle Management

Joined To Logical Interface:
System Architecture Interface
System Architecture Support

LOG.1.15 Unclassified Work Stations

Built In Higher-Level Component(s):
LOG.1 Logical Architecture: Product Lifecycle Management

Joined To Logical Interface:
Customer Unclassified Product Interface
Product Lifecycle Management Unclassified Interface
Unclassified Data Interface
Unclassified Data Interface
Unclassified Work Station Interface

7.2 Logical Architecture for Product Lifecycle Management Level 2

LOG.2 Logical Architecture: Product Lifecycle Management Department

Built In Higher-Level Component(s):
LOG.1 Logical Architecture: Product Lifecycle Management

Built From Lower-Level Component(s):
LOG.2.1 Change Advisory Boards
LOG.2.2 Business Operations Team
LOG.2.3 Design Engineering Configuration Management Team
LOG.2.4 PDMLink Business Team
LOG.2.5 Product Structure Team
LOG.2.6 Production Definition & Tools Team
LOG.2.7 Software Tool Maintenance
LOG.2.8 Systems Engineering Team
LOG.2.9 Access Control
LOG.2.10 IT Support
LOG.2.11 Product Definition
LOG.2.12 Product Lifecycle Management Strategic Planning
LOG.2.13 Product Structure
LOG.2.14 Software Change Control
LOG.2.15 Software Support
LOG.2.16 System Requirements
LOG.2.17 Management

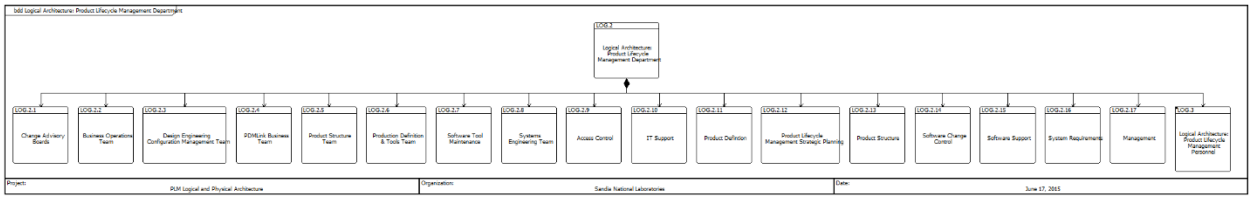


Figure 73. Block Definition of Logical Architecture Level 2

7.2.1 Internal Block Diagram for Logical Architecture Level 2

The yellow box in Figure 74 shows teams that will perform the four main services provided by the PLM.

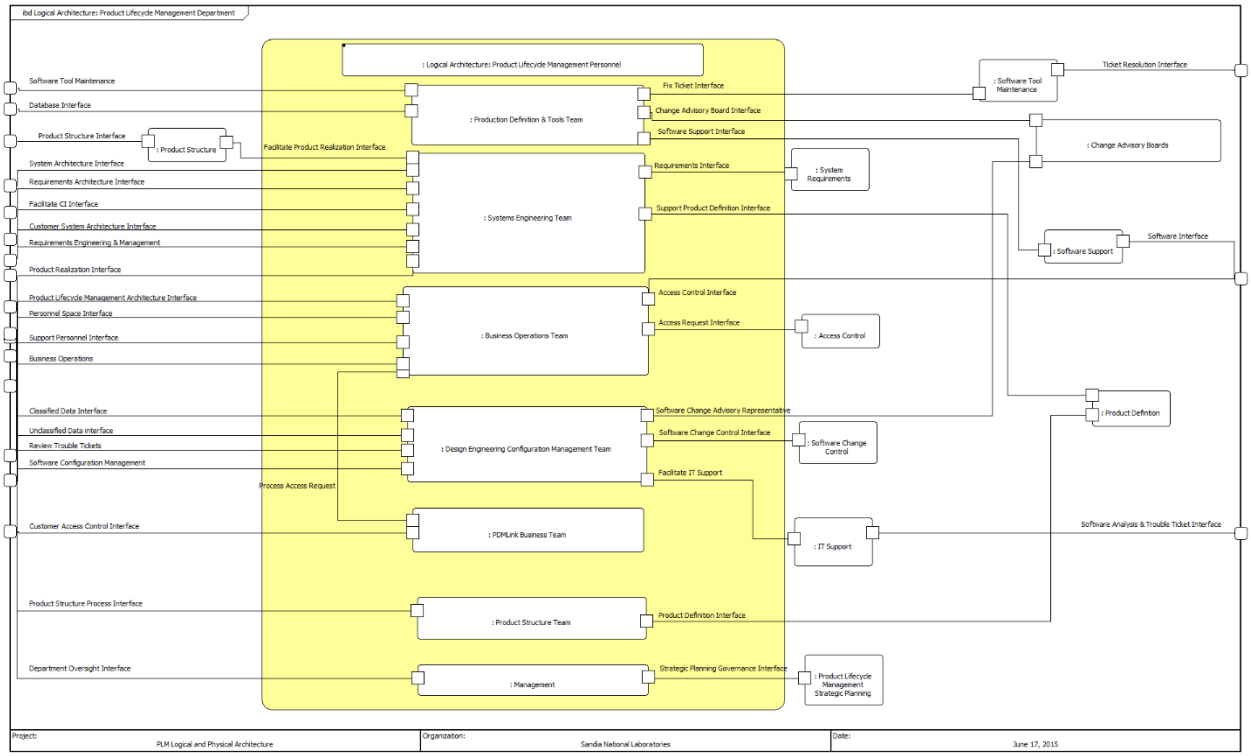


Figure 74. Internal Block Diagram for Logical Architecture Level 2

Below are the interfce details for level 2 interfaces.

LOG.2.1 Change Advisory Boards

Built In Higher-Level Component(s):
LOG.1.3 Customers
LOG.2 Logical Architecture: Product Lifecycle Management Department

Joined To Logical Interface:
Change Advisory Board Interface
Software Change Advisory Representative

Performs Logical(s):
PLMLog.2.3 Change Advisory Boards

LOG.2.1 Change Advisory Boards

Built In Higher-Level Component(s):
LOG.1.3 Customers
LOG.2 Logical Architecture: Product Lifecycle Management Department

Joined To Logical Interface:
Change Advisory Board Interface
Software Change Advisory Representative

Performs Logical(s):
PLMLog.2.3 Change Advisory Boards

LOG.2.2 Business Operations Team

Built In Higher-Level Component(s):
LOG.2 Logical Architecture: Product Lifecycle Management Department

Joined To Logical Interface:
Access Control Interface
Access Request Interface
Business Operations
Personnel Space Interface
Process Access Request
Product Lifecycle Management Architecture Interface
Support Personnel Interface

Performs Logical(s):
PLMLog.2.2 Business Operations Team

LOG.2.3 Design Engineering Configuration Management Team

Built In Higher-Level Component(s):
LOG.2 Logical Architecture: Product Lifecycle Management Department

Joined To Logical Interface:
Classified Data Interface
Facilitate IT Support
Review Trouble Tickets
Software Change Advisory Representative
Software Change Control Interface
Software Configuration Management
Unclassified Data interface

Performs Logical(s):
PLMFunc.2.4 Design Engineering Configuration Management Team

LOG.2.4 PDMLink Business Team

Built In Higher-Level Component(s):
LOG.2 Logical Architecture: Product Lifecycle Management Department

Joined To Logical Interface:
Customer Access Control Interface
Process Access Request

Performs Logical(s):
PLMLog.2.6 PDMLink Business Team

LOG.2.5 Product Structure Team

Built In Higher-Level Component(s):
LOG.2 Logical Architecture: Product Lifecycle Management Department

Joined To Logical Interface:
Product Definition Interface
Product Structure Process Interface

Performs Logical(s):
PLMLog.2.10 Product Structure Team

LOG.2.6 Production Definition & Tools Team

Built In Higher-Level Component(s):
LOG.2 Logical Architecture: Product Lifecycle Management Department

Joined To Logical Interface:
Change Advisory Board Interface
Database Interface
Fix Ticket Interface
Software Support Interface
Software Tool Maintenance

Performs Logical(s):
PLMLog.2.11 Production Definition & Tools Team

LOG.2.7 Software Tool Maintenance

Built In Higher-Level Component(s):
LOG.2 Logical Architecture: Product Lifecycle Management Department

Joined To Logical Interface:
Fix Ticket Interface
Ticket Resolution Interface

Performs Logical(s):
PLMLog.2.14 Software Tool Maintenance

LOG.2.8 Systems Engineering Team

Built In Higher-Level Component(s):
LOG.2 Logical Architecture: Product Lifecycle Management Department

Joined To Logical Interface:
Customer System Architecture Interface
Facilitate CI Interface
Facilitate Product Realization Interface
Product Realization Interface
Requirements Architecture Interface
Requirements Engineering & Management
Requirements Interface
Support Product Definition Interface

System Architecture Interface

LOG.2.9 Access Control

Built In Higher-Level Component(s):
LOG.2 Logical Architecture: Product Lifecycle Management Department

Joined To Logical Interface:
Access Request Interface

LOG.2.10 IT Support

Built In Higher-Level Component(s):
LOG.2 Logical Architecture: Product Lifecycle Management Department

Joined To Logical Interface:
Facilitate IT Support
Software Analysis & Trouble Ticket Interface

LOG.2.11 Product Definition

Built In Higher-Level Component(s):
LOG.2 Logical Architecture: Product Lifecycle Management Department

Joined To Logical Interface:
Product Definition Interface
Support Product Definition Interface

LOG.2.12 Product Lifecycle Management Strategic Planning

Built In Higher-Level Component(s):
LOG.2 Logical Architecture: Product Lifecycle Management Department

Joined To Logical Interface:
Strategic Planning Governance Interface

LOG.2.13 Product Structure

Built In Higher-Level Component(s):
LOG.2 Logical Architecture: Product Lifecycle Management Department

Joined To Logical Interface:
Facilitate Product Realization Interface
Product Structure Interface

LOG.2.14 Software Change Control

Built In Higher-Level Component(s):
LOG.2 Logical Architecture: Product Lifecycle Management Department

Joined To Logical Interface:
Software Change Control Interface

LOG.2.15 Software Support

Built In Higher-Level Component(s):
LOG.2 Logical Architecture: Product Lifecycle Management Department

Joined To Logical Interface:
Software Interface
Software Support Interface

LOG.2.16 System Requirements

Built In Higher-Level Component(s):
LOG.2 Logical Architecture: Product Lifecycle Management Department

Joined To Logical Interface:
Requirements Interface

LOG.2.17 Management

Built In Higher-Level Component(s):
LOG.2 Logical Architecture: Product Lifecycle Management Department

Joined To Logical Interface:
Department Oversight Interface
Strategic Planning Governance Interface

7.3 Logical Architecture for Product Lifecycle Management Level 3

LOG.3 Logical Architecture: Product Lifecycle Management Personnel

Built In Higher-Level Component(s):
LOG.2 Logical Architecture: Product Lifecycle Management Department

Built From Lower-Level Component(s):
LOG.3.1 Computer Aided Design & Drafting Technologist
LOG.3.2 Engineering Document Control Technologist
LOG.3.3 Engineering Program/Project Lead
LOG.3.4 Engineering Systems Integration/Implementation Pro
LOG.3.5 Information Systems Security Technologist
LOG.3.6 Manager, R&D Science & Engineering
LOG.3.7 Solutions Architect

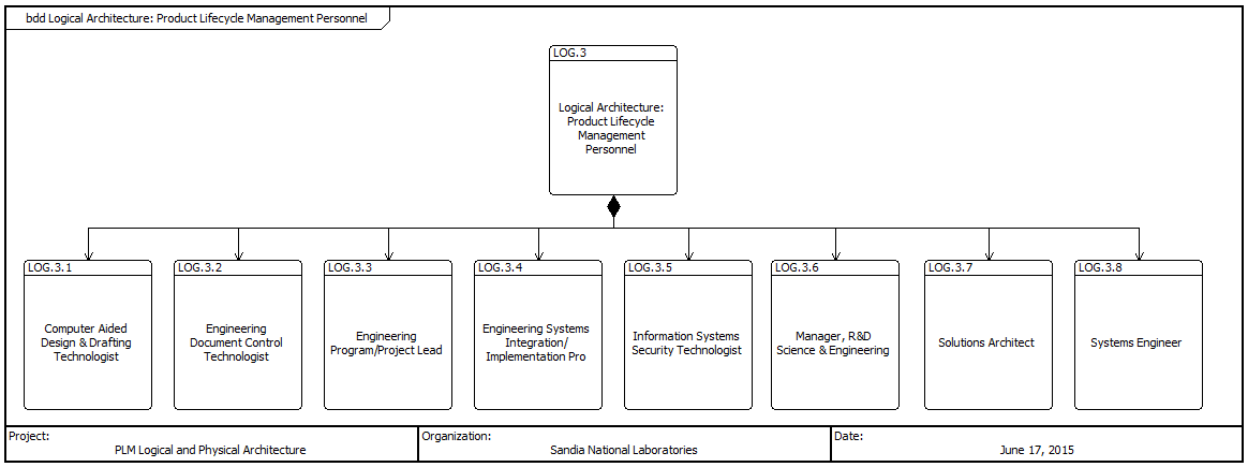


Figure 75. Block Definition of Logical Architecture Level 3

7.3.1 Internal Block Diagram for Logical Architecture Level 3

Figure 76 shows the internal block diagram for PLM personnel.

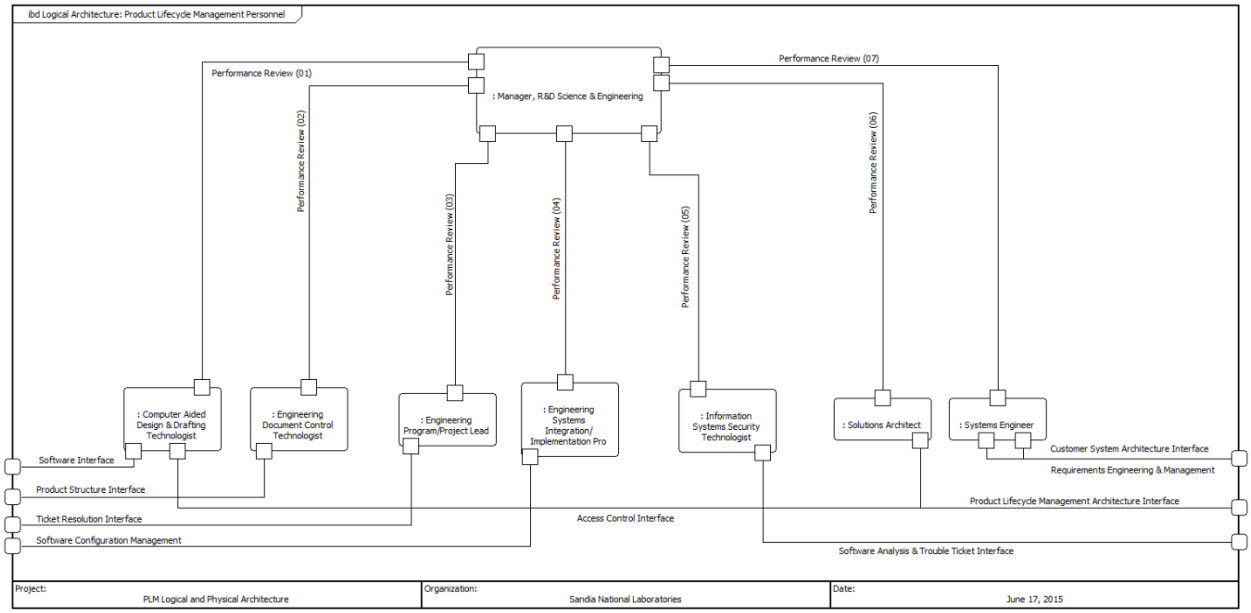


Figure 76. Internal Block Diagram for Logical Architecture Level 3

LOG.3.1 Computer Aided Design & Drafting Technologist

Built In Higher-Level Component(s):
LOG.3 Logical Architecture: Product Lifecycle Management Personnel

Joined To Logical Interface:
Access Control Interface
Performance Review (01)
Software Interface

LOG.3.2 Engineering Document Control Technologist

Built In Higher-Level Component(s):
LOG.3 Logical Architecture: Product Lifecycle Management Personnel

Joined To Logical Interface:
Performance Review (02)
Product Structure Interface

LOG.3.3 Engineering Program/Project Lead

Built In Higher-Level Component(s):
LOG.3 Logical Architecture: Product Lifecycle Management Personnel

Joined To Logical Interface:
Performance Review (03)
Ticket Resolution Interface

LOG.3.4 Engineering Systems Integration/Implementation Pro

Built In Higher-Level Component(s):
LOG.3 Logical Architecture: Product Lifecycle Management Personnel

Joined To Logical Interface:
Performance Review (04)
Software Configuration Management

LOG.3.5 Information Systems Security Technologist

Built In Higher-Level Component(s):
LOG.3 Logical Architecture: Product Lifecycle Management Personnel

Joined To Logical Interface:
Performance Review (05)
Software Analysis & Trouble Ticket Interface

Performs Logical(s):
PLMLog.3.5 Information Systems Security Technologist

LOG.3.6 Manager, R&D Science & Engineering

Built In Higher-Level Component(s):
LOG.3 Logical Architecture: Product Lifecycle Management Personnel

Joined To Logical Interface:
Performance Review (01)
Performance Review (02)
Performance Review (03)
Performance Review (04)
Performance Review (05)
Performance Review (06)
Performance Review (07)

LOG.3.7 Solutions Architect

Built In Higher-Level Component(s):
LOG.3 Logical Architecture: Product Lifecycle Management Personnel

Joined To Logical Interface:
Performance Review (06)
Product Lifecycle Management Architecture Interface

LOG.3.8 Systems Engineer

Built In Higher-Level Component(s):
LOG.3 Logical Architecture: Product Lifecycle Management Personnel

Joined To Logical Interface:
Customer System Architecture Interface
Performance Review (07)
Requirements Engineering & Management

7.4 Logical Interface List

This section lists all the interfaces for the logical architecture in alphabetic order.

Access Control Interface

Connecting Blocks:
- LOG.2.2 Business Operations Team
- LOG.3.1 Computer Aided Design & Drafting Technologist

Access Request Interface

Connecting Blocks:
- LOG.2.2 Business Operations Team
- LOG.2.9 Access Control

Audit Interface

Connecting Blocks:
- LOG.1.10 Quality Management
- LOG.2 Logical Architecture: Product Lifecycle Management Department

Business Operations

Connecting Blocks:
- LOG.1 Logical Architecture: Product Lifecycle Management
- LOG.2.2 Business Operations Team

Change Advisory Board Interface

Connecting Blocks:
- LOG.2.1 Change Advisory Boards
- LOG.2.6 Production Definition & Tools Team

Classified Data

Connecting Blocks:
- LOG.1.1 Classified Work Stations
- LOG.1.12 Servers

Classified Data Interface

Connecting Blocks:
- LOG.1.1 Classified Work Stations
- LOG.2.3 Design Engineering Configuration Management Team

Classified Work Station Interface

Connecting Blocks:
- LOG.1.1 Classified Work Stations
- LOG.1.6 Facilities

Configuration Item Support

Connecting Blocks:
- LOG.1.2 Configuration Management
- LOG.2 Logical Architecture: Product Lifecycle Management Department

Customer Access Control Interface

Connecting Blocks:
- LOG.1.3 Customers
- LOG.2.4 PDMLink Business Team

Customer Classified Product Interface

Connecting Blocks:
- LOG.1.1 Classified Work Stations
- LOG.1.3 Customers

Customer System Architecture Interface

Connecting Blocks:
- LOG.2.8 Systems Engineering Team
- LOG.3.8 Systems Engineer

Customer Unclassified Product Interface

Connecting Blocks:
- LOG.1.3 Customers
- LOG.1.15 Unclassified Work Stations

Database Interface

Connecting Blocks:
- LOG.1.4 Databases
- LOG.2.6 Production Definition & Tools Team

Database Support Interface

Connecting Blocks:
- LOG.1.4 Databases
- LOG.2 Logical Architecture: Product Lifecycle Management Department

Department Oversight Interface

Connecting Blocks:
- LOG.1 Logical Architecture: Product Lifecycle Management
- LOG.2.17 Management

Design Space Support

Connecting Blocks:
- LOG.1.5 Design Collaboration Support
- LOG.2 Logical Architecture: Product Lifecycle Management Department

Facilitate CI Interface

Connecting Blocks:
- LOG.1.2 Configuration Management
- LOG.2.8 Systems Engineering Team

Facilitate IT Support

Connecting Blocks:
- LOG.2.3 Design Engineering Configuration Management Team
- LOG.2.10 IT Support

Facilitate Product Realization Interface

Connecting Blocks:
- LOG.2.8 Systems Engineering Team
- LOG.2.13 Product Structure

Fix Ticket Interface

Connecting Blocks:
- LOG.2.6 Production Definition & Tools Team
- LOG.2.7 Software Tool Maintenance

Performance Review (01)

Connecting Blocks:
- LOG.3.1 Computer Aided Design & Drafting Technologist
- LOG.3.6 Manager, R&D Science & Engineering

Performance Review (02)

Connecting Blocks:
- LOG.3.2 Engineering Document Control Technologist
- LOG.3.6 Manager, R&D Science & Engineering

Performance Review (03)

Connecting Blocks:
- LOG.3.3 Engineering Program/Project Lead
- LOG.3.6 Manager, R&D Science & Engineering

Performance Review (04)

Connecting Blocks:
- LOG.3.4 Engineering Systems Integration/Implementation Pro
- LOG.3.6 Manager, R&D Science & Engineering

Performance Review (05)

Connecting Blocks:
- LOG.3.5 Information Systems Security Technologist
- LOG.3.6 Manager, R&D Science & Engineering

Performance Review (06)

Connecting Blocks:
- LOG.3.6 Manager, R&D Science & Engineering
- LOG.3.7 Solutions Architect

Performance Review (07)

Connecting Blocks:
- LOG.3.6 Manager, R&D Science & Engineering
- LOG.3.8 Systems Engineer

Personnel Space Interface

Connecting Blocks:
- LOG.1.6 Facilities
- LOG.2.2 Business Operations Team

Physical Space Interface

Connecting Blocks:
- LOG.1.6 Facilities
- LOG.2 Logical Architecture: Product Lifecycle Management Department

Process Access Request

Connecting Blocks:
- LOG.2.2 Business Operations Team
- LOG.2.4 PDMLink Business Team

Product Definition Interface

Connecting Blocks:
- LOG.2.5 Product Structure Team
- LOG.2.11 Product Definition

Product Definition Services

Connecting Blocks:
- LOG.1.8 Product Definition
- LOG.2 Logical Architecture: Product Lifecycle Management Department

Product Design Interface

Connecting Blocks:
- LOG.1.3 Customers
- LOG.1.7 Product

Product Lifecycle Management Architecture Interface

Connecting Blocks:
- LOG.2.2 Business Operations Team
- LOG.3.7 Solutions Architect

Product Lifecycle Management Classified Interface

Connecting Blocks:
- LOG.1.1 Classified Work Stations
- LOG.2 Logical Architecture: Product Lifecycle Management Department

Product Lifecycle Management Unclassified Interface

Connecting Blocks:
- LOG.1.15 Unclassified Work Stations
- LOG.2 Logical Architecture: Product Lifecycle Management Department

Product Realization Interface

Connecting Blocks:
- LOG.1.8 Product Definition
- LOG.2.8 Systems Engineering Team

Product Realization Support

Connecting Blocks:
- LOG.1.7 Product
- LOG.2 Logical Architecture: Product Lifecycle Management Department

Product Structure Interface

Connecting Blocks:
- LOG.2.13 Product Structure
- LOG.3.2 Engineering Document Control Technologist

Product Structure Process Interface

Connecting Blocks:
- LOG.1.8 Product Definition
- LOG.2.5 Product Structure Team

Requirements Architecture Interface

Connecting Blocks:
- LOG.1.11 Requirements Engineering & Management
- LOG.2.8 Systems Engineering Team

Requirements Architecture Support

Connecting Blocks:
- LOG.1.11 Requirements Engineering & Management
- LOG.2 Logical Architecture: Product Lifecycle Management Department

Requirements Engineering & Management

Connecting Blocks:
- LOG.2.8 Systems Engineering Team
- LOG.3.8 Systems Engineer

Requirements Interface

Connecting Blocks:
- LOG.2.8 Systems Engineering Team
- LOG.2.16 System Requirements

Review Trouble Tickets

Connecting Blocks:
- LOG.1.10.1 Tickets
- LOG.2.3 Design Engineering Configuration Management Team

Software Analysis & Trouble Ticket Interface

Connecting Blocks:
- LOG.2.10 IT Support
- LOG.3.5 Information Systems Security Technologist

Software Change Advisory Representative

Connecting Blocks:
- LOG.2.1 Change Advisory Boards
- LOG.2.3 Design Engineering Configuration Management Team

Software Change Control Interface

Connecting Blocks:
- LOG.2.3 Design Engineering Configuration Management Team
- LOG.2.14 Software Change Control

Software Configuration Management

Connecting Blocks:
- LOG.2.3 Design Engineering Configuration Management Team
- LOG.3.4 Engineering Systems Integration/Implementation Pro

Software Interface

Connecting Blocks:
- LOG.2.15 Software Support
- LOG.3.1 Computer Aided Design & Drafting Technologist

Software Support Interface

Connecting Blocks:
- LOG.2.6 Production Definition & Tools Team
- LOG.2.15 Software Support

Software Tool Maintenance

Connecting Blocks:
- LOG.1.13 Software Tools
- LOG.2.6 Production Definition & Tools Team

Software Tool Support

Connecting Blocks:
- LOG.1.13 Software Tools
- LOG.2 Logical Architecture: Product Lifecycle Management Department

Strategic Planning Governance Interface

Connecting Blocks:
- LOG.2.12 Product Lifecycle Management Strategic Planning
- LOG.2.17 Management

Support Personnel Interface

Connecting Blocks:
- LOG.1 Logical Architecture: Product Lifecycle Management
- LOG.2.2 Business Operations Team

Support Product Definition Interface

Connecting Blocks:
- LOG.2.8 Systems Engineering Team
- LOG.2.11 Product Definition

System Architecture Interface

Connecting Blocks:
- LOG.1.14 Systems Engineering Services
- LOG.2.8 Systems Engineering Team

System Architecture Support

Connecting Blocks:
- LOG.1.14 Systems Engineering Services
- LOG.2 Logical Architecture: Product Lifecycle Management Department

Ticket Resolution Interface

Connecting Blocks:
- LOG.2.7 Software Tool Maintenance
- LOG.3.3 Engineering Program/Project Lead

Unclassified Data Interface

Connecting Blocks:
- LOG.1.15 Unclassified Work Stations
- LOG.2.3 Design Engineering Configuration Management Team

Unclassified Data Interface

Connecting Blocks:
- LOG.1.12 Servers
- LOG.1.15 Unclassified Work Stations

Unclassified Work Station Interface

Connecting Blocks:
LOG.1.6 Facilities
LOG.1.15 Unclassified Work Stations

7.5 Allocation of Logical Architecture to Requirements

The allocation occurred by taking a pass of the logical architecture and determining which requirements are based on that particular block. Based on how the requirements were decomposed (section 5 and 6) there are several "sets" of requirements. For an example, Figure 77 shows a set of initiation requirements which are decomposed and based on the logical architecture. As Buede explains, "The allocated architecture integrates the requirements decomposed with the functional and physical architecture" (Buede, 2009). This section describes the first step of that process where functional is subtitled with logical. Each set of requirements is based on the logical architecture.

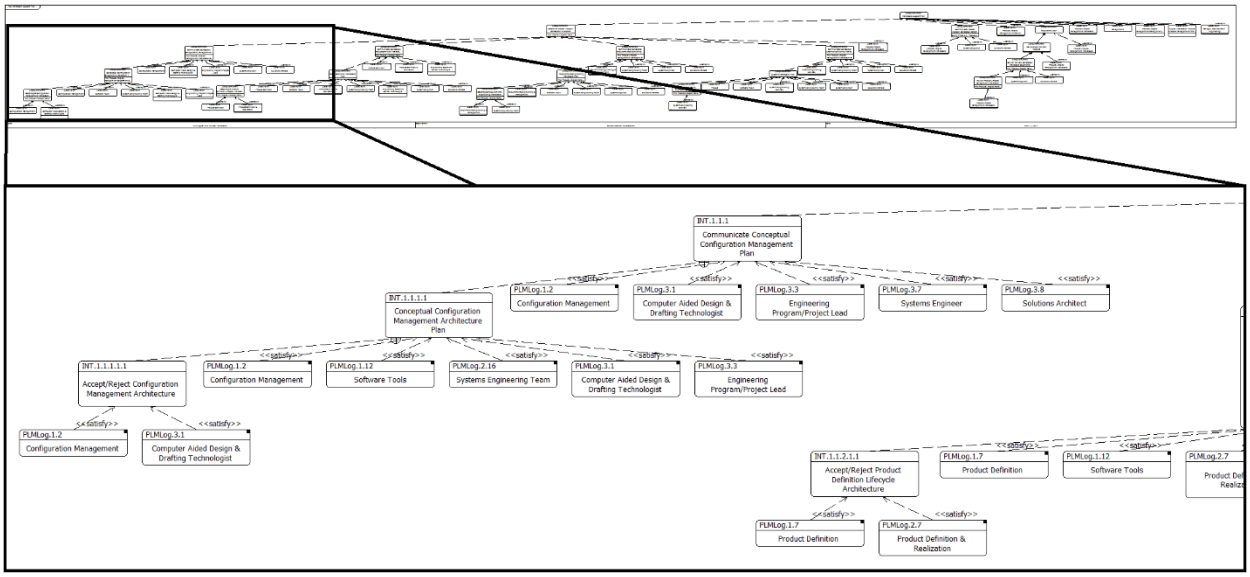


Figure 77. Example of Decomposed Requirements and Logical Allocation

7.5.1 Integration Set of Requirements and Bases of Logical Architecture

INT.1 Product Lifecycle Management Initiation

Requirement Statement:

The Product Lifecycle Management Architecture shall provide an architecture plan to support a nuclear weapon lifecycle program based on an official request from the Nuclear Weapon Strategic Management Unit (NWSMU) for a conceptual Product Lifecycle Management Initiation. At a minimum, the plan should include configuration management plan, product definition architecture, systems engineering architecture, and requirements engineering architecture no later than 30 working days after receipt of request from NWSMU.

Refined By Subordinate Requirements:

INT.1.1 Document Nuclear Weapon Conception & Request
INT.1.2 Communicate Nuclear Weapon Conceptual Design
INT.1.3 Request Revision

Basis Of:

Logical: PLMLog.1 Product Lifecycle Management Architecture
Logical: PLMLog.2 Product Lifecycle Management Personnel
Logical: PLMLog.2.5 Management
Logical: PLMLog.2.8 Product Lifecycle Management Strategic Planning
Logical: PLMLog.3 Logical Architecture: Product Lifecycle Management Personnel
Logical: PLMLog.3.6 Manager, R&D Science & Engineering

INT.1.1 Document Nuclear Weapon Conception & Request

Requirement Statement:

The Product Lifecycle Management Department shall formally document reception to develop configuration management planning, product definition architecture, systems engineering architecture, and requirements engineering architecture support within 7 calendar days of request from the NMSU.

Refines Higher-Level Requirement:

INT.1 Product Lifecycle Management Initiation

Refined By Subordinate Requirements:

INT.1.1.1 Communicate Conceptual Configuration Management Plan
INT.1.1.2 Communicate Conceptual Product Definition Architecture
INT.1.1.3 Communicate Conceptual Requirements Engineering Plan
INT.1.1.4 Communicate Conceptual Systems Engineering Architecture

Basis Of:

Logical: PLMLog.1 Product Lifecycle Management Architecture

INT.1.1.1 Communicate Conceptual Configuration Management Plan

Requirement Statement:

The Product Lifecycle Management Architecture shall communicate to the Nuclear Weapon Strategic Management Unit the conceptual configuration management plan.

Reference Paragraph Number & Title:

CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:

INT.1.1 Document Nuclear Weapon Conception & Request

Refined By Subordinate Requirements:
INT.1.1.1.1 Conceptual Configuration Management Architecture Plan

Basis Of:
Logical: PLMLog.1.2 Configuration Management
Logical: PLMLog.3.1 Computer Aided Design & Drafting Technologist
Logical: PLMLog.3.3 Engineering Program/Project Lead
Logical: PLMLog.3.7 Systems Engineer
Logical: PLMLog.3.8 Solutions Architect

INT.1.1.1.1 Conceptual Configuration Management Architecture Plan

Requirement Statement:
The Product Lifecycle Management Department shall create a conceptual configuration management plan within 60 working days based on the formal request from the Nuclear Weapon Strategic Management Unit.

Refines Higher-Level Requirement:
INT.1.1.1 Communicate Conceptual Configuration Management Plan

Refined By Subordinate Requirements:
INT.1.1.1.1.1 Accept/Reject Configuration Management Architecture

Basis Of:
Logical: PLMLog.1.2 Configuration Management
Logical: PLMLog.1.12 Software Tools
Logical: PLMLog.2.16 Systems Engineering Team
Logical: PLMLog.3.1 Computer Aided Design & Drafting Technologist
Logical: PLMLog.3.3 Engineering Program/Project Lead

INT.1.1.1.1.1 Accept/Reject Configuration Management Architecture

Requirement Statement:
The Product Lifecycle Management Architecture shall accept the decision to adopt or deny PLM configuration management.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
INT.1.1.1.1 Conceptual Configuration Management Architecture Plan

Basis Of:
Logical: PLMLog.1.2 Configuration Management
Logical: PLMLog.3.1 Computer Aided Design & Drafting Technologist

INT.1.1.2 Communicate Conceptual Product Definition Architecture

Requirement Statement:
The Product Lifecycle Management Architecture shall communicate to the Nuclear Weapon Strategic Management Unit the product definition architecture plan.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
INT.1.1 Document Nuclear Weapon Conception & Request

Refined By Subordinate Requirements:
INT.1.1.2.1 Product Definition Architecture Plan

Basis Of:
Logical: PLMLog.1.7 Product Definition
Logical: PLMLog.2.7 Product Definition & Realization
Logical: PLMLog.3.2 Engineering Document Control Technologist

INT.1.1.2.1 Product Definition Architecture Plan

Requirement Statement:
The Product Lifecycle Management Department shall create a conceptual product definition architecture plan within 30 working days based on the formal request from the Nuclear Weapon Strategic Management Unit.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
INT.1.1.2 Communicate Conceptual Product Definition Architecture

Refined By Subordinate Requirements:
INT.1.1.2.1.1 Accept/Reject Product Definition Lifecycle Architecture

Basis Of:
Logical: PLMLog.1.7 Product Definition
Logical: PLMLog.1.12 Software Tools
Logical: PLMLog.2.7 Product Definition & Realization
Logical: PLMLog.2.16 Systems Engineering Team
Logical: PLMLog.3.2 Engineering Document Control Technologist
Logical: PLMLog.3.7 Systems Engineer
Logical: PLMLog.3.8 Solutions Architect

INT.1.1.2.1.1 Accept/Reject Product Definition Lifecycle Architecture

Requirement Statement:
The Product Lifecycle Management Architecture shall accept the decision to adopt or deny PLM product structure.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
INT.1.1.2.1 Product Definition Architecture Plan

Basis Of:
Logical: PLMLog.1.7 Product Definition
Logical: PLMLog.2.7 Product Definition & Realization

INT.1.1.3 Communicate Conceptual Requirements Engineering Plan

Requirement Statement:
The Product Lifecycle Management Architecture shall communicate to the Nuclear Weapon Strategic Management Unit the requirements engineering architecture plan.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
INT.1.1 Document Nuclear Weapon Conception & Request

Refined By Subordinate Requirements:
INT.1.1.3.1 Requirements Engineering Architecture Plan

Basis Of:
Logical: PLMLog.1.10 Requirements Engineering & Management
Logical: PLMLog.2.16 Systems Engineering Team
Logical: PLMLog.3.7 Systems Engineer
Logical: PLMLog.3.8 Solutions Architect

INT.1.1.3.1 Requirements Engineering Architecture Plan

Requirement Statement:
The Product Lifecycle Management Department shall create a conceptual requirements engineering architecture plan within 30 working days based on the formal request from the Nuclear Weapon Strategic Management Unit.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
INT.1.1.3 Communicate Conceptual Requirements Engineering Plan

Refined By Subordinate Requirements:
INT.1.1.3.1.1 Accept/Reject Requirements Engineering Architecture

Basis Of:
Logical: PLMLog.1.10 Requirements Engineering & Management
Logical: PLMLog.1.12 Software Tools
Logical: PLMLog.2.16 Systems Engineering Team
Logical: PLMLog.3.7 Systems Engineer
Logical: PLMLog.3.8 Solutions Architect

INT.1.1.3.1.1 Accept/Reject Requirements Engineering Architecture

Requirement Statement:
The Product Lifecycle Management Architecture shall formally accept the decision to adopt or deny PLM requirements engineering architecture support.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
INT.1.1.3.1 Requirements Engineering Architecture Plan

Basis Of:
Logical: PLMLog.1.10 Requirements Engineering & Management
Logical: PLMLog.2.16 Systems Engineering Team

INT.1.1.4 Communicate Conceptual Systems Engineering Architecture

Requirement Statement:
The Product Lifecycle Management Architecture shall communicate to the Nuclear Weapon Strategic Management Unit the systems engineering architecture plan.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
INT.1.1 Document Nuclear Weapon Conception & Request

Refined By Subordinate Requirements:
INT.1.1.4.1 Systems Architecture Plan

Basis Of:
Logical: PLMLog.1.13 Systems Engineering Services
Logical: PLMLog.2.16 Systems Engineering Team
Logical: PLMLog.3.7 Systems Engineer
Logical: PLMLog.3.8 Solutions Architect

INT.1.1.4.1 Systems Architecture Plan

Requirement Statement:
The Product Lifecycle Management Department shall create a conceptual systems engineering architecture plan within 30 working days based on the formal request from the Nuclear Weapon Strategic Management Unit.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
INT.1.1.4 Communicate Conceptual Systems Engineering Architecture

Refined By Subordinate Requirements:
INT.1.1.4.1.1 Accept/Reject Systems Engineering Architecture

Basis Of:
Logical: PLMLog.1.8 Product
Logical: PLMLog.1.12 Software Tools
Logical: PLMLog.1.13 Systems Engineering Services
Logical: PLMLog.2.16 Systems Engineering Team
Logical: PLMLog.3.7 Systems Engineer
Logical: PLMLog.3.8 Solutions Architect

INT.1.1.4.1.1 Accept/Reject Systems Engineering Architecture

Requirement Statement:
The Product Lifecycle Management Architecture shall formally accept the decision to adopt or deny PLM systems engineering architecture support.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
INT.1.1.4.1 Systems Architecture Plan

Basis Of:
Logical: PLMLog.1.13 Systems Engineering Services

INT.1.2 Communicate Nuclear Weapon Conceptual Design

Requirement Statement:

During Phase 6.2, Product Lifecycle Management Architecture shall have a process to receive conceptual nuclear weapon scope from the NMSU within 5 working days of concept selection.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
INT.1 Product Lifecycle Management Initiation

Basis Of:
Logical: PLMLog.1 Product Lifecycle Management Architecture
Logical: PLMLog.3.7 Systems Engineer
Logical: PLMLog.3.8 Solutions Architect

INT.1.3 Request Revision

Requirement Statement:
The Product Lifecycle Management Architecture shall receive revision request from the Nuclear Weapon Strategic Management Unit.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
INT.1 Product Lifecycle Management Initiation

Refined By Subordinate Requirements:
INT.1.3.1 Acknowledge Revision Request

Basis Of:
Logical: PLMLog.1 Product Lifecycle Management Architecture

INT.1.3.1 Acknowledge Revision Request

Requirement Statement:
The Product Lifecycle Management Architecture shall acknowledge all revision requests from the Nuclear Weapon Strategic Management Unit within 5 working days.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
INT.1.3 Request Revision

Refined By Subordinate Requirements:
INT.1.3.1.1 Deliver Revised Conceptual Architecture Documents

Basis Of:
Logical: PLMLog.1 Product Lifecycle Management Architecture

INT.1.3.1.1.1 Review Product Lifecycle Management Architecture

Requirement Statement:
The Product Lifecycle Management Architecture shall review the program architecture plan with the Nuclear Weapon Strategic Management Unit within 15 calendar days after completion of INT.1.1.1.1, INT.1.1.2.1, INT.1.1.3.1 and INT.1.1.4.1.

Reference Paragraph Number & Title:

CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
INT.1.3.1.1 Deliver Revised Conceptual Architecture Documents

Basis Of:
Logical: PLMLog.1 Product Lifecycle Management Architecture

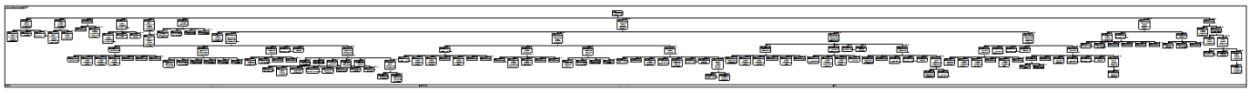


Figure 78. Allocation of Initiation Set of Requirements to Logical Architecture

7.5.2 Execution Set of Requirements and Bases of Logical Architecture

EX.1.1 Configuration Management Items

Requirement Statement:
The Product Lifecycle Management Architecture shall receive a configuration management item list from the Nuclear Weapon program.

Refines Higher-Level Requirement:
EX.1 Product Lifecycle Management Execution

Refined By Subordinate Requirements:
EX.1.1.1 Implement Configuration Management

Basis Of:
Logical: PLMFunc.2.4 Design Engineering Configuration Management Team
Logical: PLMLog.1.3 Customers
Logical: PLMLog.2.16 Systems Engineering Team
Logical: PLMLog.3.7 Systems Engineer
Logical: PLMLog.3.8 Solutions Architect

EX.1.1.1 Implement Configuration Management

Requirement Statement:
The Product Lifecycle Management Architecture shall implement configuration management architecture within 90 working days of authorization to proceed to 6.3.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
EX.1.1 Configuration Management Items

Basis Of:
Logical: PLMFunc.2.4 Design Engineering Configuration Management Team
Logical: PLMLog.1.3 Customers
Logical: PLMLog.1.5 Design Collaboration Support
Logical: PLMLog.2.16 Systems Engineering Team
Logical: PLMLog.3.1 Computer Aided Design & Drafting Technologist
Logical: PLMLog.3.7 Systems Engineer
Logical: PLMLog.3.8 Solutions Architect

EX.1.2 Product Definition Item

Requirement Statement:
The Product Lifecycle Management Architecture shall receive product definition data from the Nuclear Weapon program.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
EX.1 Product Lifecycle Management Execution

Refined By Subordinate Requirements:
EX.1.2.1 Implement Product Definition Architecture

Basis Of:
Logical: PLMLog.1.3 Customers
Logical: PLMLog.1.7 Product Definition
Logical: PLMLog.2.7 Product Definition & Realization
Logical: PLMLog.2.9 Product Structure
Logical: PLMLog.2.10 Product Structure Team
Logical: PLMLog.2.16 Systems Engineering Team
Logical: PLMLog.3.2 Engineering Document Control Technologist
Logical: PLMLog.3.7 Systems Engineer
Logical: PLMLog.3.8 Solutions Architect

EX.1.2.1 Implement Product Definition Architecture

Requirement Statement:
The Product Lifecycle Management Architecture shall implement product definition architecture within 90 working days of authorization to proceed to 6.3.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
EX.1.2 Product Definition Item

Basis Of:
Logical: PLMLog.1.3 Customers
Logical: PLMLog.1.5 Design Collaboration Support
Logical: PLMLog.2.7 Product Definition & Realization
Logical: PLMLog.2.10 Product Structure Team
Logical: PLMLog.2.16 Systems Engineering Team
Logical: PLMLog.3.2 Engineering Document Control Technologist
Logical: PLMLog.3.7 Systems Engineer
Logical: PLMLog.3.8 Solutions Architect

EX.1.3 System Architecture

Requirement Statement:
The Product Lifecycle Management Architecture shall have a process for receiving a system concept of operations from the Nuclear Weapon Program.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
EX.1 Product Lifecycle Management Execution

Refined By Subordinate Requirements:
EX.1.3.1 Implement System Architecture

Basis Of:
Logical: PLMLog.1.13 Systems Engineering Services
Logical: PLMLog.2.15 System Requirements
Logical: PLMLog.2.16 Systems Engineering Team
Logical: PLMLog.3.7 Systems Engineer
Logical: PLMLog.3.8 Solutions Architect

EX.1.3.1 Implement System Architecture

Requirement Statement:
The Product Lifecycle Management Architecture shall implement baseline system architecture for the system under design within 90 working days of authorization to proceed to 6.3.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
EX.1.3 System Architecture

Basis Of:
Logical: PLMLog.1.3 Customers
Logical: PLMLog.1.5 Design Collaboration Support
Logical: PLMLog.1.8 Product
Logical: PLMLog.1.13 Systems Engineering Services
Logical: PLMLog.2.16 Systems Engineering Team
Logical: PLMLog.3.7 Systems Engineer
Logical: PLMLog.3.8 Solutions Architect

EX.1.4 Customer Requirements Support

Requirement Statement:
Upon request from the customer, the Product Lifecycle Management Architecture shall deploy a requirements team to support nuclear weapon design requirements.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
EX.1 Product Lifecycle Management Execution

Refined By Subordinate Requirements:
EX.1.4.1 Requirements Engineering Architecture

Basis Of:
Logical: PLMLog.1.3 Customers
Logical: PLMLog.1.6 Facilities
Logical: PLMLog.1.10 Requirements Engineering & Management
Logical: PLMLog.2.15 System Requirements
Logical: PLMLog.2.16 Systems Engineering Team
Logical: PLMLog.3.7 Systems Engineer

Logical: PLMLog.3.8 Solutions Architect

EX.1.4.1 Requirements Engineering Architecture

Requirement Statement:
The Product Lifecycle Management Architecture shall receive customer (originating) requirements from the Nuclear Weapon program.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
EX.1.4 Customer Requirements Support

Basis Of:
Logical: PLMLog.1.3 Customers
Logical: PLMLog.1.5 Design Collaboration Support
Logical: PLMLog.1.10 Requirements Engineering & Management
Logical: PLMLog.2.15 System Requirements
Logical: PLMLog.3.7 Systems Engineer
Logical: PLMLog.3.8 Solutions Architect

EX.1.5 Accept Access Request

Requirement Statement:
The Product Lifecycle Management Architecture shall have a documented process for accepting access requests from the Nuclear Weapon Program regarding Need to know information residing within nuclear weapon development software tools.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
EX.1 Product Lifecycle Management Execution

Refined By Subordinate Requirements:
EX.1.5.1 Manage Access Request

Basis Of:
Logical: PLMLog.1.3 Customers
Logical: PLMLog.2.1 Access Control
Logical: PLMLog.2.6 PDMLink Business Team
Logical: PLMLog.3.1 Computer Aided Design & Drafting Technologist
Logical: PLMLog.3.5 Information Systems Security Technologist

EX.1.5.1 Manage Access Request

Requirement Statement:
The Product Lifecycle Management Architecture shall produce access control reports upon request in common format. Common format is defined as Microsoft Excel Spreadsheets, Microsoft Power Point Presentations, Microsoft Word, PDF, HTML, and Visio.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
EX.1.5 Accept Access Request

Basis Of:
Logical: PLMLog.1.14 Tickets
Logical: PLMLog.2.1 Access Control
Logical: PLMLog.2.2 Business Operations Team
Logical: PLMLog.2.6 PDMLink Business Team
Logical: PLMLog.3.1 Computer Aided Design & Drafting Technologist

EX.1.7 Product Lifecycle Management IT Support

Requirement Statement:
The Product Lifecycle Management Architecture shall have a documented process for supporting IT requests from the nuclear weapon customer.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
EX.1 Product Lifecycle Management Execution

Basis Of:
Logical: PLMLog.1.1 Classified Work Stations
Logical: PLMLog.1.3 Customers
Logical: PLMLog.1.4 Databases
Logical: PLMLog.1.5 Design Collaboration Support
Logical: PLMLog.1.11 Servers
Logical: PLMLog.1.12 Software Tools
Logical: PLMLog.1.14 Tickets
Logical: PLMLog.1.15 Unclassified Work Stations
Logical: PLMLog.2.4 Perform IT Support
Logical: PLMLog.2.11 Production Definition & Tools Team
Logical: PLMLog.2.12 Software Change Control
Logical: PLMLog.2.13 Software Support
Logical: PLMLog.2.14 Software Tool Maintenance
Logical: PLMLog.3.3 Engineering Program/Project Lead
Logical: PLMLog.3.4 Engineering Systems Integration/Implementation Pro
Logical: PLMLog.3.5 Information Systems Security Technologist

EX.1.8 Product Structure Evolution Process

Requirement Statement:
The Product Lifecycle Management Architecture shall have a documented process for establishing traceability from product evolution to product realization.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
EX.1 Product Lifecycle Management Execution

Refined By Subordinate Requirements:
EX.1.8.1 Change Advisory Board

Basis Of:
Logical: PLMLog.1.7 Product Definition
Logical: PLMLog.3.3 Engineering Program/Project Lead

EX.1.8.1 Change Advisory Board

Requirement Statement:
The Product Lifecycle Management Architecture shall have an established change advisory board which serves as the overarching change advisory board.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
EX.1.8 Product Structure Evolution Process

Basis Of:
Logical: PLMLog.1.3 Customers
Logical: PLMLog.2.2 Business Operations Team
Logical: PLMLog.2.3 Change Advisory Boards
Logical: PLMLog.2.4 Perform IT Support
Logical: PLMLog.2.6 PDMLink Business Team
Logical: PLMLog.2.12 Software Change Control
Logical: PLMLog.3.3 Engineering Program/Project Lead
Logical: PLMLog.3.4 Engineering Systems Integration/Implementation Pro
Logical: PLMLog.3.5 Information Systems Security Technologist

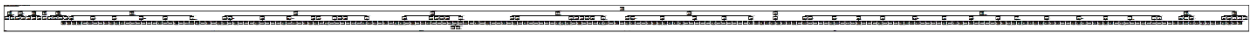


Figure 79. Allocation of Execution Set of Requirements to Logical Architecture

7.5.3 Monitoring & Controlling Set of Requirements and Bases of Logical Architecture

MC.1.1 Production Customer Support

Requirement Statement:
The Product Lifecycle Management Architecture shall provide production support to include configuration management, product structure, and systems engineering.

Refines Higher-Level Requirement:
MC.1 Product Lifecycle Management Architecture Monitoring & Controlling

Refined By Subordinate Requirements:
MC.1.1.1 Production Product Structure Support
MC.1.1.2 Production Configuration Management
MC.1.1.3 Production Systems Engineering Support

Basis Of:
Logical: PLMLog.1.3 Customers
Logical: PLMLog.2.7 Product Definition & Realization
Logical: PLMLog.3.7 Systems Engineer
Logical: PLMLog.3.8 Solutions Architect

MC.1.1.1.1 Production Product Realization

Requirement Statement:
The Product Lifecycle Management Architecture shall have a process for revising part defining definition changes.

Reference Paragraph Number & Title:

CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
MC.1.1.1 Production Product Structure Support

Basis Of:
Logical: PLMLog.1.8 Product
Logical: PLMLog.2.7 Product Definition & Realization
Logical: PLMLog.3.1 Computer Aided Design & Drafting Technologist
Logical: PLMLog.3.2 Engineering Document Control Technologist

MC.1.1.2.1 Manage Production Changes

Requirement Statement:
The Product Lifecycle Management Architecture shall have a process for managing production change control & configuration items.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
MC.1.1.2 Production Configuration Management

Basis Of:
Logical: PLMLog.1.2 Configuration Management
Logical: PLMLog.2.3 Change Advisory Boards

MC.1.1.3 Production Systems Engineering Support

Requirement Statement:
The Product Lifecycle Management Architecture shall provide production systems engineering support which includes requirements & system architecture support.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
MC.1.1 Production Customer Support

Refined By Subordinate Requirements:
MC.1.1.3.1 Requirements & Architecture Support

Basis Of:
Logical: PLMLog.1.8 Product
Logical: PLMLog.1.10 Requirements Engineering & Management
Logical: PLMLog.1.13 Systems Engineering Services
Logical: PLMLog.2.15 System Requirements
Logical: PLMLog.2.16 Systems Engineering Team
Logical: PLMLog.3.7 Systems Engineer
Logical: PLMLog.3.8 Solutions Architect

MC.1.1.3.1 Requirements & Architecture Support

Requirement Statement:
The Product Lifecycle Management Architecture shall have a process for supporting system integration, verification, & validation.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
MC.1.1.3 Production Systems Engineering Support

Basis Of:
Logical: PLMLog.1.8 Product
Logical: PLMLog.1.10 Requirements Engineering & Management
Logical: PLMLog.1.13 Systems Engineering Services
Logical: PLMLog.2.15 System Requirements
Logical: PLMLog.2.16 Systems Engineering Team
Logical: PLMLog.3.7 Systems Engineer
Logical: PLMLog.3.8 Solutions Architect

MC.1.2 Production Software/Hardware/IT Support

Requirement Statement:
The Product Lifecycle Management Architecture shall provide production software, hardware, and IT support.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
MC.1 Product Lifecycle Management Architecture Monitoring & Controlling

Refined By Subordinate Requirements:
MC.1.2.1 Change Advisory Boards

Basis Of:
Logical: PLMLog.1.11 Servers
Logical: PLMLog.1.12 Software Tools
Logical: PLMLog.2.12 Software Change Control
Logical: PLMLog.3.1 Computer Aided Design & Drafting Technologist
Logical: PLMLog.3.3 Engineering Program/Project Lead
Logical: PLMLog.3.4 Engineering Systems Integration/Implementation Pro
Logical: PLMLog.3.5 Information Systems Security Technologist

MC.1.2.1 Production Product Definition

Requirement Statement:
The Product Lifecycle Management Architecture shall provide production product definition support for software, hardware, and IT support through 5 change advisory boards.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
MC.1.2.1 Change Advisory Boards

Basis Of:
Logical: PLMLog.1.2 Configuration Management
Logical: PLMLog.1.7 Product Definition
Logical: PLMLog.2.3 Change Advisory Boards
Logical: PLMLog.2.4 Perform IT Support

Logical: PLMLog.2.13 Software Support

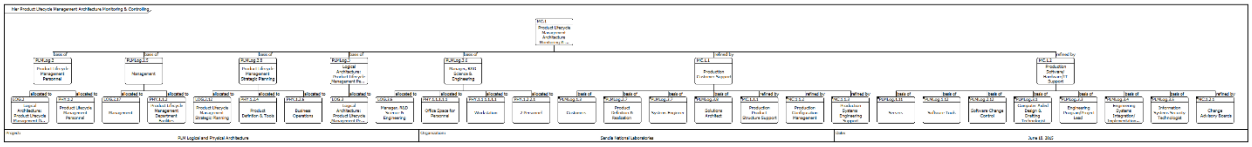


Figure 80. Allocation of Monitoring & Controlling Set of Requirements to Logical Architecture

7.5.4 Quality Control Set of Requirements and Bases of Logical Architecture

QC.1 Product Lifecycle Management Quality Control

Requirement Statement:

The Product Lifecycle Management Architecture shall provide quality control plans to the nuclear weapon program during phase 6.5 which includes maintenance quality control & customer quality control.

Refined By Subordinate Requirements:

QC.1.1 First Production Maintenance Quality Control
QC.1.2 First Production Customer Quality Control

Basis Of:

Logical: PLMLog.1.3 Customers
Logical: PLMLog.1.9 Quality Management
Logical: PLMLog.2 Product Lifecycle Management Personnel
Logical: PLMLog.2.5 Management
Logical: PLMLog.2.8 Product Lifecycle Management Strategic Planning
Logical: PLMLog.3 Logical Architecture: Product Lifecycle Management Personnel
Logical: PLMLog.3.5 Information Systems Security Technologist
Logical: PLMLog.3.6 Manager, R&D Science & Engineering
Logical: PLMLog.3.7 Systems Engineer
Logical: PLMLog.3.8 Solutions Architect

QC.1.1.1.1 Audit Ticketing Process

Requirement Statement:

The Product Lifecycle Management Architecture shall conduct internal audits to include the following items:
1) Categories of tickets
2) Number of tickets in each category
3) Process for determining ticket priority and technical impact analysis
4) Average time for resolution in each category
5) Resulting tickets generated from any resolution

Reference Paragraph Number & Title:

CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:

QC.1.1.1 First Production Product Definition Maintenance Quality Control

Basis Of:

Logical: PLMLog.1.12 Software Tools
Logical: PLMLog.1.14 Tickets

Logical: PLMLog.2.2 Business Operations Team
Logical: PLMLog.3.3 Engineering Program/Project Lead
Logical: PLMLog.3.4 Engineering Systems Integration/Implementation Pro

QC.1.2.1 Audit Configuration Management Architecture

Requirement Statement:
The Product Lifecycle Management Architecture shall provide first production quality control which includes a process for auditing configuration management items and processes.

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
QC.1.2 First Production Customer Quality Control

Refined By Subordinate Requirements:
QC.1.2.1.1 Perform Configuration Management Audits

Basis Of:
Logical: PLMLog.1.12 Software Tools
Logical: PLMLog.2.2 Business Operations Team
Logical: PLMLog.2.12 Software Change Control
Logical: PLMLog.3.3 Engineering Program/Project Lead
Logical: PLMLog.3.4 Engineering Systems Integration/Implementation Pro

QC.1.2.1.1 Perform Configuration Management Audits

Requirement Statement:
The Product Lifecycle Management Architecture shall have a process for performing configuration management audits which include the following:
1) T-30 Documentation
2) Change control process for development work
3) Change control process for released work

Reference Paragraph Number & Title:
CL.1.1.1.2 Retirement & Record Archiving

Refines Higher-Level Requirement:
QC.1.2.1 Audit Configuration Management Architecture

Basis Of:
Logical: PLMLog.1.12 Software Tools
Logical: PLMLog.2.2 Business Operations Team
Logical: PLMLog.3.1 Computer Aided Design & Drafting Technologist
Logical: PLMLog.3.3 Engineering Program/Project Lead
Logical: PLMLog.3.4 Engineering Systems Integration/Implementation Pro

Figure 81. Allocation of Quality Control Set of Requirements to Logical Architecture

7.5.5 Closing Set of Requirements and Bases of Logical Architecture

CL.1 Product Lifecycle Management Architecture Closing Support

Requirement Statement:

Upon reception of dismantlement support request, the Product Lifecycle Management Architecture shall respond to the Nuclear Weapon Program with formal communication the intended dismantlement support architecture within 60 working days.

Reference Paragraph Number & Title:

CL.1 Product Lifecycle Management Architecture Closing Support

Refined By Subordinate Requirements:

CL.1.1 Product Definition & Configuration Management

Basis Of:

Function: PLMLog.1.3 Customers
Function: PLMLog.2 Product Lifecycle Management Personnel
Function: PLMLog.2.5 Management
Function: PLMLog.2.8 Product Lifecycle Management Strategic Planning
Function: PLMLog.3 Logical Architecture: Product Lifecycle Management Personnel
Function: PLMLog.3.6 Manager, R&D Science & Engineering
Function: PLMLog.3.7 Systems Engineer
Function: PLMLog.3.8 Solutions Architect

CL.1.1.1.1 Material List/Procurement Index Maintenance

Requirement Statement:

The Product Lifecycle Management Architecture shall have a process for maintaining closing material list and procurement index maintenance support.

Reference Paragraph Number & Title:

CL.1.1.1.1 Material List/Procurement Index Maintenance

Refines Higher-Level Requirement:

CL.1.1.1 Legacy Part Definition Maintenance & Closing Support

Basis Of:

Function: PLMLog.1.14 Tickets
Function: PLMLog.3.2 Engineering Document Control Technologist

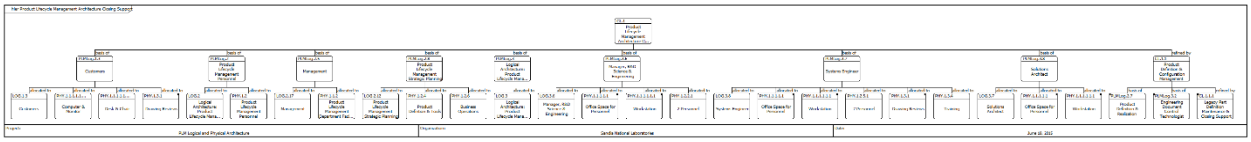


Figure 82. Allocation of Closing Set of Requirements to Logical Architecture

7.6 Physical Architecture Product Lifecycle Management

As Buede describes, “The physical architecture of a system is a hierarchical description of the resources that comprise the system. This hierarchy begins with the system and the system’s top-level components and progresses down to the configuration items (CIs) that comprise each

intermediate component. The CIs can be hardware, software blocks or combinations of hardware and software, people, facilities, procedures, and documents (e.g., user manuals)" (Buede, 2009). This section details the physical architecture of the PLM.

7.6.1 Physical Architecture Hierarchy List

PHY.1 Product Lifecycle Management
- PHY.1.1 Product Lifecycle Management Facilities
  - PHY.1.1.1 Deployed Product Lifecycle Management Personnel
    - PHY.1.1.1.1 Customer Space
      - PHY.1.1.1.1.1 Office Space for Personnel
        - PHY.1.1.1.1.1.1 Workstation
          - PHY.1.1.1.1.1.1.1 Computer & Monitor
            - PHY.1.1.1.1.1.1.1.1 Unclassified Environment
            - PHY.1.1.1.1.1.1.1.2 Classified Environment
          - PHY.1.1.1.1.1.1.2 Desk & Chair
  - PHY.1.1.2 Product Lifecycle Management Department Facilities
    - PHY.1.1.2.1 Product Lifecycle Management Space
      - PHY.1.1.1.1.1 Office Space for Personnel
      - PHY.1.1.2.1.1 Meeting/Collaboration Space
        - PHY.1.1.2.1.1.1 Meeting Table & Chairs
          - PHY.1.1.2.1.1.1.1 Computer & Projector
  - PHY.1.1.3 Server Facilities
    - PHY.1.1.3.1 HVAC
    - PHY.1.1.3.2 Server Space
- PHY.1.2 Product Lifecycle Management Personnel
  - PHY.1.2.1 Design Engineering Configuration Management
    - PHY.1.2.1.1 5 Personnel
  - PHY.1.2.2 Manager
    - PHY.1.2.2.1 2 Personnel
  - PHY.1.2.3 Product Definition & Realization
    - PHY.1.2.3.1 12 Personnel
  - PHY.1.2.4 Product Definition & Tools
    - PHY.1.2.4.1 6 Personnel
  - PHY.1.2.5 System Engineering
    - PHY.1.2.5.1 7 Personnel
  - PHY.1.2.6 Business Operations
    - PHY.1.2.6.1 4 Personnel
- PHY.1.3 Product Lifecycle Management Services
  - PHY.1.3.1 Drawing Reviews
  - PHY.1.3.2 Hardware

PHY.1.3.3 Software

PHY.1.3.4 Training

PHY.1.4 Product Lifecycle Management Tools

PHY.1.4.1 Computers

PHY.1.4.2 Pen & Report

PHY.1.5 Customer Product

PHY.1.5.1 Physical Product

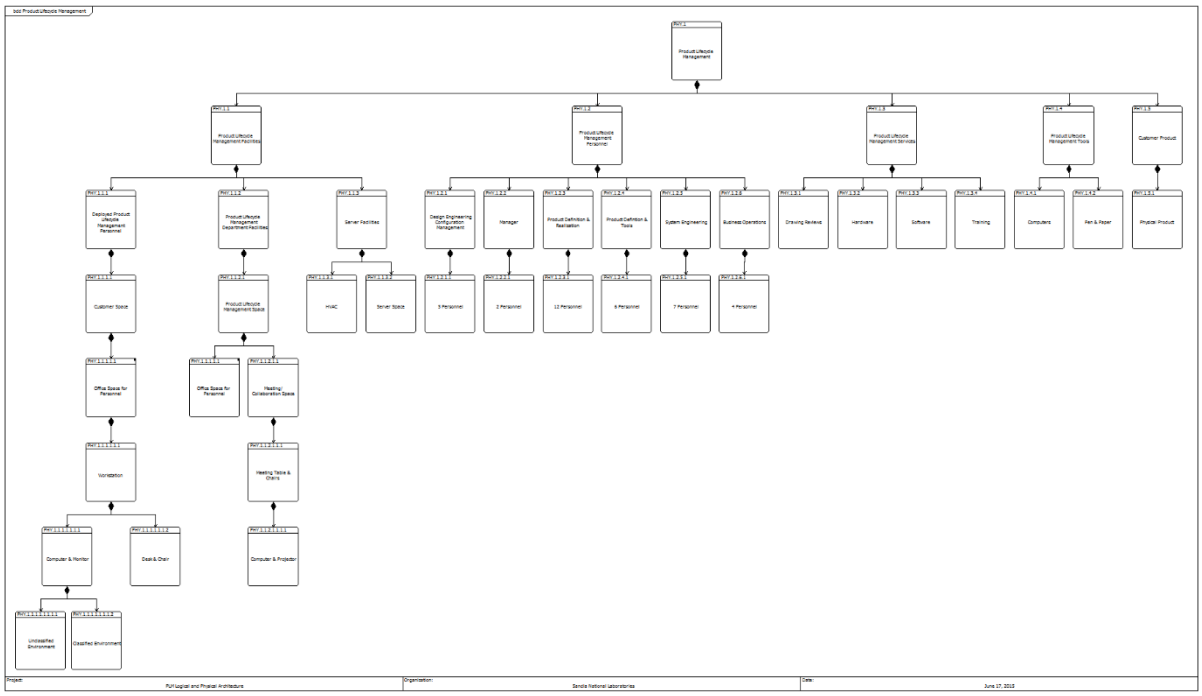


Figure 83. Physical Architecture of the Product Lifecycle Management

7.6.2 Logical Architecture Allocated to the Physical Architecture

This section lists the allocation from logical architecture (PLMLog) to physical architecture (PHY). When describing the allocation of logical to physical, a relationship of "allocated to" exists. An example of how the logical blocks are allocated to the physical blocks is displayed in Figure 84 below. In the example, the logical element of facilities (PLMLog.1.6) is allocated to a list of physical blocks.

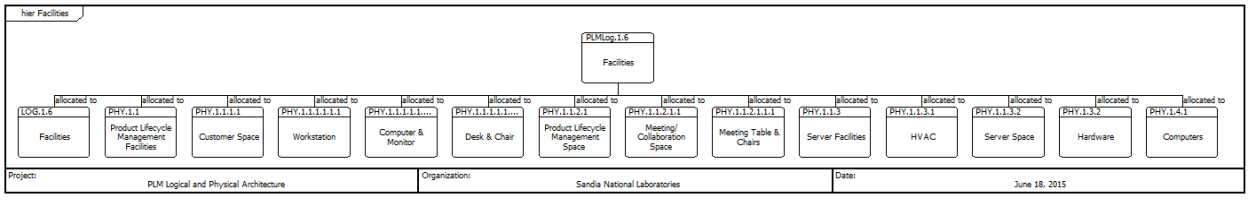


Figure 84. Example of Logical Architecture to Physical Architecture

When reading the relationship of physical to logical, a relationship of *"performs"* exists. The performs relationship from physical to logical is the detail contained in this section.

PHY.1 Product Lifecycle Management

Built From Lower-Level Component(s):
- PHY.1.1 Product Lifecycle Management Facilities
- PHY.1.2 Product Lifecycle Management Personnel
- PHY.1.3 Product Lifecycle Management Services
- PHY.1.4 Product Lifecycle Management Tools
- PHY.1.5 Customer Product

PHY.1.1 Product Lifecycle Management Facilities

Built In Higher-Level Component(s):
- PHY.1 Product Lifecycle Management

Built From Lower-Level Component(s):
- PHY.1.1.1 Deployed Product Lifecycle Management Personnel
- PHY.1.1.2 Product Lifecycle Management Department Facilities
- PHY.1.1.3 Server Facilities

Performs Logical(s):
- PLMLog.1.1 Classified Work Stations
- PLMLog.1.2 Configuration Management
- PLMLog.1.4 Databases
- PLMLog.1.5 Design Collaboration Support
- PLMLog.1.6 Facilities
- PLMLog.1.7 Product Definition
- PLMLog.1.9 Quality Management
- PLMLog.1.10 Requirements Engineering & Management
- PLMLog.1.11 Servers
- PLMLog.1.12 Software Tools
- PLMLog.1.13 Systems Engineering Services
- PLMLog.1.14 Tickets
- PLMLog.1.15 Unclassified Work Stations

PHY.1.1.1 Deployed Product Lifecycle Management Personnel

Built In Higher-Level Component(s):
- PHY.1.1 Product Lifecycle Management Facilities

Built From Lower-Level Component(s):
- PHY.1.1.1.1 Customer Space

PHY.1.1.1.1 Customer Space

Built In Higher-Level Component(s):
- PHY.1.1.1 Deployed Product Lifecycle Management Personnel

Built From Lower-Level Component(s):
- PHY.1.1.1.1.1 Office Space for Personnel

Performs Logical(s):

PLMLog.1.6 Facilities

PHY.1.1.1.1.1 Office Space for Personnel

Built In Higher-Level Component(s):
PHY.1.1.1.1 Customer Space
PHY.1.1.2.1 Product Lifecycle Management Space

Built From Lower-Level Component(s):
PHY.1.1.1.1.1.1 Workstation

Performs Logical(s):
PLMLog.3.1 Computer Aided Design & Drafting Technologist
PLMLog.3.2 Engineering Document Control Technologist
PLMLog.3.3 Engineering Program/Project Lead
PLMLog.3.4 Engineering Systems Integration/Implementation Pro
PLMLog.3.5 Information Systems Security Technologist
PLMLog.3.6 Manager, R&D Science & Engineering
PLMLog.3.7 Systems Engineer
PLMLog.3.8 Solutions Architect

PHY.1.1.1.1.1.1 Workstation

Built In Higher-Level Component(s):
PHY.1.1.1.1.1 Office Space for Personnel

Built From Lower-Level Component(s):
PHY.1.1.1.1.1.1.1 Computer & Monitor
PHY.1.1.1.1.1.1.2 Desk & Chair

Performs Logical(s):
PLMLog.1.1 Classified Work Stations
PLMLog.1.6 Facilities
PLMLog.1.15 Unclassified Work Stations
PLMLog.3.1 Computer Aided Design & Drafting Technologist
PLMLog.3.2 Engineering Document Control Technologist
PLMLog.3.3 Engineering Program/Project Lead
PLMLog.3.4 Engineering Systems Integration/Implementation Pro
PLMLog.3.5 Information Systems Security Technologist
PLMLog.3.6 Manager, R&D Science & Engineering
PLMLog.3.7 Systems Engineer
PLMLog.3.8 Solutions Architect

PHY.1.1.1.1.1.1.1 Computer & Monitor

Built In Higher-Level Component(s):
PHY.1.1.1.1.1.1 Workstation

Built From Lower-Level Component(s):
PHY.1.1.1.1.1.1.1.1 Unclassified Environment
PHY.1.1.1.1.1.1.1.2 Classified Environment

Performs Logical(s):
PLMLog.1.3 Customers
PLMLog.1.4 Databases

PLMLog.1.6 Facilities

PHY.1.1.1.1.1.1.1.1 Unclassified Environment

Built In Higher-Level Component(s):
PHY.1.1.1.1.1.1.1 Computer & Monitor

Performs Logical(s):
PLMLog.1.12 Software Tools
PLMLog.1.15 Unclassified Work Stations

PHY.1.1.1.1.1.1.1.2 Classified Environment

Built In Higher-Level Component(s):
PHY.1.1.1.1.1.1.1 Computer & Monitor

Performs Logical(s):
PLMLog.1.1 Classified Work Stations
PLMLog.1.12 Software Tools

PHY.1.1.1.1.1.1.2 Desk & Chair

Built In Higher-Level Component(s):
PHY.1.1.1.1.1.1 Workstation

Performs Logical(s):
PLMLog.1.3 Customers
PLMLog.1.6 Facilities

PHY.1.1.2 Product Lifecycle Management Department Facilities

Built In Higher-Level Component(s):
PHY.1.1 Product Lifecycle Management Facilities

Built From Lower-Level Component(s):
PHY.1.1.2.1 Product Lifecycle Management Space

Performs Logical(s):
PLMFunc.2.4 Design Engineering Configuration Management Team
PLMLog.2.1 Access Control
PLMLog.2.2 Business Operations Team
PLMLog.2.3 Change Advisory Boards
PLMLog.2.4 Perform IT Support
PLMLog.2.5 Management
PLMLog.2.6 PDMLink Business Team
PLMLog.2.7 Product Definition & Realization
PLMLog.2.9 Product Structure
PLMLog.2.10 Product Structure Team
PLMLog.2.11 Production Definition & Tools Team
PLMLog.2.12 Software Change Control
PLMLog.2.13 Software Support
PLMLog.2.14 Software Tool Maintenance
PLMLog.2.16 Systems Engineering Team

PHY.1.1.2.1 Product Lifecycle Management Space

Built In Higher-Level Component(s):
PHY.1.1.2 Product Lifecycle Management Department Facilities

Built From Lower-Level Component(s):
PHY.1.1.1.1.1 Office Space for Personnel
PHY.1.1.2.1.1 Meeting/Collaboration Space

Performs Logical(s):
PLMFunc.2.4 Design Engineering Configuration Management Team
PLMLog.1.6 Facilities
PLMLog.2.2 Business Operations Team
PLMLog.2.6 PDMLink Business Team
PLMLog.2.10 Product Structure Team
PLMLog.2.11 Production Definition & Tools Team
PLMLog.2.16 Systems Engineering Team

PHY.1.1.1.1.1 Office Space for Personnel

Built In Higher-Level Component(s):
PHY.1.1.1.1 Customer Space
PHY.1.1.2.1 Product Lifecycle Management Space

Built From Lower-Level Component(s):
PHY.1.1.1.1.1.1 Workstation

Performs Logical(s):
PLMLog.3.1 Computer Aided Design & Drafting Technologist
PLMLog.3.2 Engineering Document Control Technologist
PLMLog.3.3 Engineering Program/Project Lead
PLMLog.3.4 Engineering Systems Integration/Implementation Pro
PLMLog.3.5 Information Systems Security Technologist
PLMLog.3.6 Manager, R&D Science & Engineering
PLMLog.3.7 Systems Engineer
PLMLog.3.8 Solutions Architect

PHY.1.1.2.1.1 Meeting/Collaboration Space

Built In Higher-Level Component(s):
PHY.1.1.2.1 Product Lifecycle Management Space

Built From Lower-Level Component(s):
PHY.1.1.2.1.1.1 Meeting Table & Chairs

Performs Logical(s):
PLMLog.1.6 Facilities

PHY.1.1.2.1.1.1 Meeting Table & Chairs

Built In Higher-Level Component(s):
PHY.1.1.2.1.1 Meeting/Collaboration Space

Built From Lower-Level Component(s):
PHY.1.1.2.1.1.1.1 Computer & Projector

Performs Logical(s):
PLMLog.1.6 Facilities

PHY.1.1.2.1.1.1.1 Computer & Projector

Built In Higher-Level Component(s):
PHY.1.1.2.1.1.1 Meeting Table & Chairs

Performs Logical(s):
PLMLog.1.12 Software Tools

PHY.1.1.3 Server Facilities

Built In Higher-Level Component(s):
PHY.1.1 Product Lifecycle Management Facilities

Built From Lower-Level Component(s):
PHY.1.1.3.1 HVAC
PHY.1.1.3.2 Server Space

Performs Logical(s):
PLMLog.1.6 Facilities

PHY.1.1.3.1 HVAC

Built In Higher-Level Component(s):
PHY.1.1.3 Server Facilities

Performs Logical(s):
PLMLog.1.6 Facilities

PHY.1.1.3.2 Server Space

Built In Higher-Level Component(s):
PHY.1.1.3 Server Facilities

Performs Logical(s):
PLMLog.1.6 Facilities

PHY.1.2 Product Lifecycle Management Personnel

Built In Higher-Level Component(s):
PHY.1 Product Lifecycle Management

Built From Lower-Level Component(s):
PHY.1.2.1 Design Engineering Configuration Management
PHY.1.2.2 Manager
PHY.1.2.3 Product Definition & Realization
PHY.1.2.4 Product Definition & Tools
PHY.1.2.5 System Engineering
PHY.1.2.6 Business Operations

Performs Logical(s):
PLMLog.2 Product Lifecycle Management Personnel

PHY.1.2.1 Design Engineering Configuration Management

Built In Higher-Level Component(s):
PHY.1.2 Product Lifecycle Management Personnel

Built From Lower-Level Component(s):
PHY.1.2.1.1 5 Personnel

Performs Logical(s):
PLMFunc.2.4 Design Engineering Configuration Management Team
PLMLog.2.3 Change Advisory Boards
PLMLog.2.6 PDMLink Business Team
PLMLog.2.12 Software Change Control
PLMLog.2.13 Software Support
PLMLog.2.14 Software Tool Maintenance

PHY.1.2.1.1 5 Personnel

Built In Higher-Level Component(s):
PHY.1.2.1 Design Engineering Configuration Management

Performs Logical(s):
PLMLog.3.1 Computer Aided Design & Drafting Technologist

PHY.1.2.2 Manager

Built In Higher-Level Component(s):
PHY.1.2 Product Lifecycle Management Personnel

Built From Lower-Level Component(s):
PHY.1.2.2.1 2 Personnel

Performs Logical(s):
PLMLog.1 Product Lifecycle Management Architecture

PHY.1.2.2.1 2 Personnel

Built In Higher-Level Component(s):
PHY.1.2.2 Manager

Performs Logical(s):
PLMLog.3.6 Manager, R&D Science & Engineering

PHY.1.2.3 Product Definition & Realization

Built In Higher-Level Component(s):
PHY.1.2 Product Lifecycle Management Personnel

Built From Lower-Level Component(s):
PHY.1.2.3.1 12 Personnel

Performs Logical(s):
PLMLog.2.7 Product Definition & Realization
PLMLog.2.9 Product Structure
PLMLog.2.10 Product Structure Team

PHY.1.2.3.1 12 Personnel

Built In Higher-Level Component(s):
PHY.1.2.3 Product Definition & Realization

Performs Logical(s):
PLMLog.3.1 Computer Aided Design & Drafting Technologist

PHY.1.2.4 Product Definition & Tools

Built In Higher-Level Component(s):
PHY.1.2 Product Lifecycle Management Personnel

Built From Lower-Level Component(s):
PHY.1.2.4.1 6 Personnel

Performs Logical(s):
PLMLog.2.3 Change Advisory Boards
PLMLog.2.4 Perform IT Support
PLMLog.2.6 PDMLink Business Team
PLMLog.2.8 Product Lifecycle Management Strategic Planning
PLMLog.2.11 Production Definition & Tools Team
PLMLog.2.12 Software Change Control
PLMLog.2.13 Software Support
PLMLog.2.14 Software Tool Maintenance

PHY.1.2.4.1 6 Personnel

Built In Higher-Level Component(s):
PHY.1.2.4 Product Definition & Tools

Performs Logical(s):
PLMLog.3.4 Engineering Systems Integration/Implementation Pro
PLMLog.3.5 Information Systems Security Technologist

PHY.1.2.5 System Engineering

Built In Higher-Level Component(s):
PHY.1.2 Product Lifecycle Management Personnel

Built From Lower-Level Component(s):
PHY.1.2.5.1 7 Personnel

Performs Logical(s):
PLMLog.2.9 Product Structure
PLMLog.2.12 Software Change Control
PLMLog.2.15 System Requirements
PLMLog.2.16 Systems Engineering Team

PHY.1.2.5.1 7 Personnel

Built In Higher-Level Component(s):
PHY.1.2.5 System Engineering

Performs Logical(s):
PLMLog.3.7 Systems Engineer

PHY.1.2.6 Business Operations

Built In Higher-Level Component(s):
PHY.1.2 Product Lifecycle Management Personnel

Built From Lower-Level Component(s):
PHY.1.2.6.1 4 Personnel

Performs Logical(s):
PLMLog.2.1 Access Control
PLMLog.2.2 Business Operations Team
PLMLog.2.3 Change Advisory Boards
PLMLog.2.6 PDMLink Business Team
PLMLog.2.8 Product Lifecycle Management Strategic Planning
PLMLog.2.12 Software Change Control

PHY.1.2.6.1 4 Personnel

Built In Higher-Level Component(s):
PHY.1.2.6 Business Operations

Performs Logical(s):
PLMLog.3.3 Engineering Program/Project Lead

PHY.1.3 Product Lifecycle Management Services

Built In Higher-Level Component(s):
PHY.1 Product Lifecycle Management

Built From Lower-Level Component(s):
PHY.1.3.1 Drawing Reviews
PHY.1.3.2 Hardware
PHY.1.3.3 Software
PHY.1.3.4 Training

Performs Logical(s):
PLMLog.1.2 Configuration Management
PLMLog.1.7 Product Definition
PLMLog.1.10 Requirements Engineering & Management
PLMLog.1.13 Systems Engineering Services

PHY.1.3.1 Drawing Reviews

Built In Higher-Level Component(s):
PHY.1.3 Product Lifecycle Management Services

Performs Logical(s):
PLMLog.1.3 Customers
PLMLog.1.8 Product
PLMLog.1.12 Software Tools
PLMLog.3.7 Systems Engineer

PHY.1.3.2 Hardware

Built In Higher-Level Component(s):
PHY.1.3 Product Lifecycle Management Services

Performs Logical(s):
PLMLog.1.1 Classified Work Stations
PLMLog.1.5 Design Collaboration Support
PLMLog.1.6 Facilities
PLMLog.1.12 Software Tools
PLMLog.1.15 Unclassified Work Stations

PHY.1.3.3 Software

Built In Higher-Level Component(s):
PHY.1.3 Product Lifecycle Management Services

Performs Logical(s):
PLMLog.1.1 Classified Work Stations
PLMLog.1.4 Databases
PLMLog.1.7 Product Definition
PLMLog.1.9 Quality Management
PLMLog.1.10 Requirements Engineering & Management
PLMLog.1.11 Servers
PLMLog.1.12 Software Tools
PLMLog.1.13 Systems Engineering Services
PLMLog.1.14 Tickets
PLMLog.1.15 Unclassified Work Stations
PLMLog.2.1 Access Control
PLMLog.2.7 Product Definition & Realization
PLMLog.2.9 Product Structure
PLMLog.2.12 Software Change Control
PLMLog.2.13 Software Support
PLMLog.2.14 Software Tool Maintenance
PLMLog.2.15 System Requirements

PHY.1.3.4 Training

Built In Higher-Level Component(s):
PHY.1.3 Product Lifecycle Management Services

Performs Logical(s):
PLMLog.3.1 Computer Aided Design & Drafting Technologist
PLMLog.3.2 Engineering Document Control Technologist
PLMLog.3.4 Engineering Systems Integration/Implementation Pro
PLMLog.3.5 Information Systems Security Technologist
PLMLog.3.7 Systems Engineer

PHY.1.4 Product Lifecycle Management Tools

Built In Higher-Level Component(s):
PHY.1 Product Lifecycle Management

Built From Lower-Level Component(s):
PHY.1.4.1 Computers
PHY.1.4.2 Pen & Report

Performs Logical(s):
PLMLog.1.2 Configuration Management

PLMLog.1.7 Product Definition
PLMLog.1.10 Requirements Engineering & Management
PLMLog.1.13 Systems Engineering Services

PHY.1.4.1 Computers

Built In Higher-Level Component(s):
PHY.1.4 Product Lifecycle Management Tools

Performs Logical(s):
PLMLog.1.1 Classified Work Stations
PLMLog.1.2 Configuration Management
PLMLog.1.6 Facilities
PLMLog.1.11 Servers
PLMLog.1.15 Unclassified Work Stations
PLMLog.2.1 Access Control

PHY.1.4.2 Pen & Report

Built In Higher-Level Component(s):
PHY.1.4 Product Lifecycle Management Tools

Performs Logical(s):
PLMLog.1.2 Configuration Management
PLMLog.1.7 Product Definition
PLMLog.2.7 Product Definition & Realization
PLMLog.2.15 System Requirements

PHY.1.5 Customer Product

Built In Higher-Level Component(s):
PHY.1 Product Lifecycle Management

Built From Lower-Level Component(s):
PHY.1.5.1 Physical Product

Performs Logical(s):
PLMLog.1 Product Lifecycle Management Architecture

PHY.1.5.1 Physical Product

Built In Higher-Level Component(s):
PHY.1.5 Customer Product

Performs Logical(s):
PLMLog.1.2 Configuration Management
PLMLog.1.5 Design Collaboration Support
PLMLog.1.7 Product Definition
PLMLog.1.10 Requirements Engineering & Management
PLMLog.1.13 Systems Engineering Services

8 FINDINGS, RECOMMENDATIONS, AND CONCLUSION

Throughout the creation of this report, there have been several topics worthy of listing as a "finding". A finding is considered a topic which may not have been previously considered relevant to the PLM. A topic which is considered a finding is offered further analysis by this report into its viability as an area of responsibility of the PLM. In addition to findings, this report contains some recommendations. Recommendations are considerations listed as steps for the PLM to take in order to become a full lifecycle service provider.

8.1 Findings

Based on the need statement in Section 3.2, this section contains a list of findings which were captured during the analysis.

- In order to fully provide a service to combat problems encountered in the need statement, the PLM must be engaged early in any program lifecycle
- Using MBSE for each program will help with traceability of design decisions to production and deployment of a product
- Requirements can be used to understand the allocation of architecting of a program and a programs system
- Through this analysis, four main services appeared as key gaps in program capabilities
    - Configuration Management
    - Product Definition and Realization
    - Systems Architecting
    - Requirements Engineering
- Customer changes during a program lifecycle can be analyzed using MBSE resulting in impacts analysis
- Interestingly, this is not the first attempt to capture a business process
    - SAND 2007-3192

8.2 Recommendations

This report recommends that the need statement in Section 3.2 be considered enough of a customer issue that the PLM create the capability to take complete ownership of such an issue early in any program lifecycle. It is recommended that the PLM engage as early as possible in

any program and provide some core capabilities. Core capabilities recommended in this report are:

- Configuration Management
- Product Definition and Realization
- Systems Architecting
- Requirements Engineering

In addition to providing core capabilities, this report also recommends that the PLM establish a set of baseline requirements. If the PLM had a baseline set of requirements, they could be used to help direct change advisory boards (CAB). Having a PLM CAB would allow for the establishment of an organized change control process and resulting in efficient resolution.

8.3 Conclusion

Section 1.2 uses Clark and Henderson's article to explain architectural innovation by presenting the need for mapping the NW 6.X process and the PLM. The analysis of this report demonstrates how MBSE is used to solve a particular problem which is built around the need statement of Section 3.2. Based on the need statement, an analysis of the architecture of both the NW 6.X Product Lifecycle and the PLM was done. Illustration of both the NW 6.X ConOps and PLM were represented with a series SysML diagrams which show various views of the model. The solution space contains an overview of stakeholders, methodology for developing PLM requirements, internal blocks of the logical architecture, the allocation of requirements to both logical and physical architecture, and an RTM in Appendix A. The real challenge for creating the PLM architecture came during the allocation of requirements. The allocation section is where the real design of the PLM architecture is described and for the intent of this report, is open for further argument. By creating a model of the PLM, issues identified by a customer can now be the basis for creating a set of requirements and solutions. Using the NW 6.X process and current capabilities of the PLM, a gap in service to customers has been analyzed and a robust solution has been developed. The solution includes configuration management services, product definition and realization, requirements engineering and management, and system architecting. All requirements developed in this report can be traced back to the highest level of the architecture and can serve as evidence of a solution. The objective is for customers to choose which services they would like to incorporate into early program planning and which services

they would like to leave out. For services left out, the PLM is able to fully demonstrate what gaps the customer is choosing to internally cover.

Using MBSE and System Modeling Language (SysML), this report has analyzed both the ConOps for the NW 6.X process and the PLM. In addition to the ConOps, further analysis covered in-depth description of the PLM architecture using a series of use cases, developed PLM requirements, logical and physical architecture, and allocation of requirements.

9 REFERENCES & RESOURCES

Blanchard, Benjamin S. *Systems Engineering Management.* Hoboken, NJ: John Wiley & Sons, Inc., 2008.

Buede, M. Dennis. *The Engineering Design of Systems Models and Methods.* Hoboken, New Jersey.: John Wiley & Sons, Inc., 2009.

Delligatti, Lenny. *SysML Distilled A Brief Guide to The Systems Modeling Language.* Upper Saddle River, NJ. : Person Education, Inc. , 2014.

Friedenthal, Sandord, Alan Moore, and Rick Steiner. *A Practical Guide to SysML.* Burlington, MA 01803: Morgan Kaufmann, 2008.

Henderson, Rebecca M., and Kim B. Clark, . *Architectural Innovation: The Reconfiguratio nof Existing Product Technologies and the Failure of Estalished Firms.* Harvard Business School: Cornell University, 2001.

Larson, Willey J., Doug Kirkpatrick, Jerry J. Sellers, Dale L. Thomas, and Dinesh Verma. *Applied Space Systems Engineering.* Hoboken, NJ: Mc Graw Hill, 2009.

SE Handbook Working Group International Council on Systems Engineering (INCOSE). *Systems Engineering Handbook.* San Diego, CA: INCOSE, 2011.

Trauth, Sharon L. *Operational Concepts and Implementation Strategies for the Design Configuraiton Management Process.* Albuquerque, New Mexico: Sandia National Laboratories SAND2007-3192, 2007.

[Blank page following section.]

10 DISTRIBUTION

Dr. Robert Cloutier
Stevens Institute of Technology
Babibo 503
Hoboken, NJ 07030

4 Lawrence Livermore National Laboratory
Attn: N. Dunipace (1)
P.O. Box 808, MS L-795
Livermore, CA 94551-0808

0624	Ty Christy	02982
0624	Tim Wiseley	02982
0472	Danny Thomas	02137

1	MS0899	Technical Library	9536 (electronic copy)

Sandia National Laboratories

Appendix A - Requirements Traceability Matrix (RTM)

Allocated Capabilities/Requirements	Traced From Higher-Level Blocks
LOG.1 Logical Architecture: Product Lifecycle Management (Component)	
PLMLog.1.1 Classified Work Stations (Logical Architecture)	EX.1.7 Product Lifecycle Management IT Support (Requirement) NWP.1.7.1 Computer Network Capabilities (Requirement)
PLMLog.1.2 Configuration Management (Logical Architecture)	INT.1.1.1 Communicate Conceptual Configuration Management Plan (Requirement) MC.1.2.1 Production Product Definition (Requirement) NWP.1.8 Product Structure Support (Requirement) NWP.1.4 Configuration Management Support (Requirement) INT.1.1.1.1 Conceptual Configuration Management Architecture Plan (Requirement) INT.1.1.1.1.1 Accept/Reject Configuration Management Architecture (Requirement) INTER.2 System Interface (Requirement) MC.1.1.2.1 Manage Production Changes (Requirement) INTER.1 Product Structure (Requirement)
PLMLog.1.3 Customers (Logical Architecture)	CL.1 Product Lifecycle Management Architecture Closing Support (Requirement) EX.1.3.1 Implement System Architecture (Requirement) EX.1.4.1 Requirements Engineering Architecture (Requirement) NWP.1.7.1 Computer Network Capabilities (Requirement) NWP.1.2 Systems Engineering Support (Requirement) QC.1 Product Lifecycle Management Quality Control (Requirement) MC.1.1 Production Customer Support (Requirement) NWP.1.1 Support Databases (Requirement) EX.1.5 Accept Access Request (Requirement) NWP.1.4 Configuration Management Support (Requirement) EX.1.8.1 Change Advisory Board (Requirement) NWP.1.3 Requirements Engineering & Management Support (Requirement) NWP.1.5 Accessing Nuclear Weapon Need (Requirement) EX.1.2 Product Definition Item (Requirement) EX.1.2.1 Implement Product Definition Architecture

Allocated Capabilities/Requirements	Traced From Higher-Level Blocks
	(Requirement) EX.1.1.1 Implement Configuration Management (Requirement) EX.1.4 Customer Requirements Support (Requirement) EX.1.7 Product Lifecycle Management IT Support (Requirement) NWP.1.6 Computer Aided Drafting & Design Support (Requirement) EX.1.1 Configuration Management Items (Requirement) NWP.1.7 Establishing Customer Computer Network Support (Requirement)
PLMLog.1.4 Databases (Logical Architecture)	INTER.2 System Interface (Requirement) EX.1.7 Product Lifecycle Management IT Support (Requirement) NWP.1.1 Support Databases (Requirement)
PLMLog.1.5 Design Collaboration Support (Logical Architecture)	EX.1.3.1 Implement System Architecture (Requirement) EX.1.7 Product Lifecycle Management IT Support (Requirement) EX.1.4.1 Requirements Engineering Architecture (Requirement) EX.1.1.1 Implement Configuration Management (Requirement) EX.1.2.1 Implement Product Definition Architecture (Requirement) NWP.1.6 Computer Aided Drafting & Design Support (Requirement) NWP.1.5 Accessing Nuclear Weapon Need (Requirement)
PLMLog.1.6 Facilities (Logical Architecture)	EX.1.4 Customer Requirements Support (Requirement)
PLMLog.1.7 Product Definition (Logical Architecture)	EX.1.8 Product Structure Evolution Process (Requirement) INT.1.1.2.1.1 Accept/Reject Product Definition Lifecycle Architecture (Requirement) MC.1.2.1 Production Product Definition (Requirement) NWP.1.8 Product Structure Support (Requirement) EX.1.2 Product Definition Item (Requirement) INT.1.1.2 Communicate Conceptual Product Definition Architecture (Requirement) INTER.2 System Interface (Requirement) INTER.1 Product Structure (Requirement) INT.1.1.2.1 Product Definition Architecture Plan (Requirement)
PLMLog.1.8 Product (Logical Architecture)	INT.1.1.4.1 Systems Architecture Plan (Requirement) EX.1.3.1 Implement System Architecture

Allocated Capabilities/Requirements	Traced From Higher-Level Blocks
	(Requirement) MC.1.1.3 Production Systems Engineering Support (Requirement) NWP.1.8 Product Structure Support (Requirement) MC.1.1.1.1 Production Product Realization (Requirement) MC.1.1.3.1 Requirements & Architecture Support (Requirement)
PLMLog.1.9 Quality Management (Logical Architecture)	QC.1 Product Lifecycle Management Quality Control (Requirement)
PLMLog.1.10 Requirements Engineering & Management (Logical Architecture)	INT.1.1.3 Communicate Conceptual Requirements Engineering Plan (Requirement) INT.1.1.3.1 Requirements Engineering Architecture Plan (Requirement) MC.1.1.3 Production Systems Engineering Support (Requirement) EX.1.4.1 Requirements Engineering Architecture (Requirement) NWP.1.3 Requirements Engineering & Management Support (Requirement) MC.1.1.3.1 Requirements & Architecture Support (Requirement) INT.1.1.3.1.1 Accept/Reject Requirements Engineering Architecture (Requirement) EX.1.4 Customer Requirements Support (Requirement)
PLMLog.1.11 Servers (Logical Architecture)	MC.1.2 Production Software/Hardware/IT Support (Requirement) EX.1.7 Product Lifecycle Management IT Support (Requirement) NWP.1.7 Establishing Customer Computer Network Support (Requirement) NWP.1.1 Support Databases (Requirement) NWP.1.7.1 Computer Network Capabilities (Requirement)
PLMLog.1.12 Software Tools (Logical Architecture)	QC.1.2.1 Audit Configuration Management Architecture (Requirement) INTER.2 System Interface (Requirement) NWP.1.7.1 Computer Network Capabilities (Requirement) MC.1.2 Production Software/Hardware/IT Support (Requirement) INT.1.1.1.1 Conceptual Configuration Management Architecture Plan (Requirement) QC.1.1.1.1 Audit Ticketing Process (Requirement) INTER.3 Tool Usability (Requirement) INT.1.1.2.1 Product Definition Architecture Plan

Allocated Capabilities/Requirements	Traced From Higher-Level Blocks
	(Requirement) INT.1.1.4.1 Systems Architecture Plan (Requirement) QC.1.2.1.1 Perform Configuration Management Audits (Requirement) EX.1.7 Product Lifecycle Management IT Support (Requirement) NWP.1.6 Computer Aided Drafting & Design Support (Requirement) INT.1.1.3.1 Requirements Engineering Architecture Plan (Requirement)
PLMLog.1.13 Systems Engineering Services (Logical Architecture)	INT.1.1.4.1.1 Accept/Reject Systems Engineering Architecture (Requirement) EX.1.3 System Architecture (Requirement) EX.1.3.1 Implement System Architecture (Requirement) INT.1.1.4 Communicate Conceptual Systems Engineering Architecture (Requirement) INT.1.1.4.1 Systems Architecture Plan (Requirement) MC.1.1.3 Production Systems Engineering Support (Requirement) MC.1.1.3.1 Requirements & Architecture Support (Requirement)
PLMLog.1.14 Tickets (Logical Architecture)	EX.1.7 Product Lifecycle Management IT Support (Requirement) EX.1.5.1 Manage Access Request (Requirement) CL.1.1.1.1 Material List/Procurement Index Maintenance (Requirement) QC.1.1.1.1 Audit Ticketing Process (Requirement)
PLMLog.1.15 Unclassified Work Stations (Logical Architecture)	EX.1.7 Product Lifecycle Management IT Support (Requirement) NWP.1.7.1 Computer Network Capabilities (Requirement)
PLMLog.1.5 Design Collaboration Support (Logical Architecture)	EX.1.3.1 Implement System Architecture (Requirement) EX.1.7 Product Lifecycle Management IT Support (Requirement) EX.1.4.1 Requirements Engineering Architecture (Requirement) EX.1.1.1 Implement Configuration Management (Requirement) EX.1.2.1 Implement Product Definition Architecture (Requirement) NWP.1.6 Computer Aided Drafting & Design Support (Requirement) NWP.1.5 Accessing Nuclear Weapon Need (Requirement)

Allocated Capabilities/Requirements	Traced From Higher-Level Blocks
LOG.1.14 Systems Engineering Services (Component)	
LOG.1.15 Unclassified Work Stations (Component)	
LOG.2 Logical Architecture: Product Lifecycle Management Department (Component)	
PLMLog.2.1 Access Control (Logical Architecture)	NWP.1.5 Accessing Nuclear Weapon Need (Requirement) EX.1.5 Accept Access Request (Requirement) EX.1.5.1 Manage Access Request (Requirement)
PLMLog.2.2 Business Operations Team (Logical Architecture)	EX.1.6 Engineering Authorization (Requirement) EX.1.8.1 Change Advisory Board (Requirement) QC.1.1.1.1 Audit Ticketing Process (Requirement) EX.1.9 Product Lifecycle Management Change Process (Requirement) MC.1.2.1 Change Advisory Boards (Requirement) QC.1.2.2.1 Perform Models & Drawing Audits (Requirement) EX.1.5.1 Manage Access Request (Requirement) QC.1.2.1.1 Perform Configuration Management Audits (Requirement) QC.1.2.1 Audit Configuration Management Architecture (Requirement)
PLMLog.2.3 Change Advisory Boards (Logical Architecture)	MC.1.2.1 Production Product Definition (Requirement) EX.1.8.1 Change Advisory Board (Requirement) EX.1.9 Product Lifecycle Management Change Process (Requirement) MC.1.1.2.1 Manage Production Changes (Requirement)
PLMLog.2.4 Perform IT Support (Logical Architecture)	EX.1.7 Product Lifecycle Management IT Support (Requirement) EX.1.8.1 Change Advisory Board (Requirement) MC.1.2.1 Production Product Definition (Requirement)
PLMLog.2.5 Management (Logical Architecture)	CL.1 Product Lifecycle Management Architecture Closing Support (Requirement) NWP.1 Product Lifecycle Management Architecture Authorization (Requirement) MC.1 Product Lifecycle Management Architecture Monitoring & Controlling (Requirement) EX.1 Product Lifecycle Management Execution(Requirement) QC.1 Product Lifecycle Management Quality Control (Requirement) INT.1 Product Lifecycle Management Initiation (Requirement)
PLMLog.2.6 PDMLink Business Team (Logical	EX.1.8.1 Change Advisory Board (Requirement)

Allocated Capabilities/Requirements	Traced From Higher-Level Blocks
Architecture)	EX.1.5 Accept Access Request (Requirement) EX.1.5.1 Manage Access Request (Requirement)
PLMLog.2.7 Product Definition & Realization (Logical Architecture)	NWP.1.8 Product Structure Support (Requirement) QC.1.2.2.1 Perform Models & Drawing Audits (Requirement) QC.1.2.2 Audit Product Production Definition Architecture (Requirement) CL.1.1 Product Definition & Configuration Management (Requirement) CL.1.1.1 Legacy Part Definition Maintenance & Closing Support (Requirement) MC.1.1 Production Customer Support (Requirement) MC.1.1.1.1 Production Product Realization (Requirement) EX.1.2 Product Definition Item (Requirement) INT.1.1.2 Communicate Conceptual Product Definition Architecture (Requirement) EX.1.2.1 Implement Product Definition Architecture (Requirement) INT.1.1.2.1 Product Definition Architecture Plan (Requirement) MC.1.1.1 Production Product Structure Support (Requirement) INT.1.1.2.1.1 Accept/Reject Product Definition Lifecycle Architecture (Requirement)
PLMLog.2.8 Product Lifecycle Management Strategic Planning (Logical Architecture)	CL.1 Product Lifecycle Management Architecture Closing Support (Requirement) NWP.1 Product Lifecycle Management Architecture Authorization (Requirement) MC.1 Product Lifecycle Management Architecture Monitoring & Controlling (Requirement) EX.1 Product Lifecycle Management Execution(Requirement) QC.1 Product Lifecycle Management Quality Control (Requirement) INT.1 Product Lifecycle Management Initiation (Requirement)
PLMLog.2.9 Product Structure (Logical Architecture)	EX.1.2 Product Definition Item (Requirement)
PLMLog.2.10 Product Structure Team (Logical Architecture)	EX.1.2 Product Definition Item (Requirement) EX.1.2.1 Implement Product Definition Architecture (Requirement)
PLMLog.2.11 Production Definition & Tools Team (Logical Architecture)	EX.1.7 Product Lifecycle Management IT Support (Requirement)
PLMLog.2.12 Software Change Control (Logical Architecture)	MC.1.2 Production Software/Hardware/IT Support (Requirement) EX.1.7 Product Lifecycle Management IT Support

Allocated Capabilities/Requirements	Traced From Higher-Level Blocks
	(Requirement) EX.1.8.1 Change Advisory Board (Requirement) NWP.1.6 Computer Aided Drafting & Design Support (Requirement) QC.1.2.1 Audit Configuration Management Architecture (Requirement)
PLMLog.2.13 Software Support (Logical Architecture)	EX.1.7 Product Lifecycle Management IT Support (Requirement) NWP.1.1 Support Databases (Requirement) MC.1.2.1 Production Product Definition (Requirement)
PLMLog.2.14 Software Tool Maintenance (Logical Architecture)	EX.1.7 Product Lifecycle Management IT Support (Requirement)
PLMLog.2.15 System Requirements (Logical Architecture)	EX.1.3 System Architecture (Requirement) MC.1.1.3 Production Systems Engineering Support (Requirement) NWP.1.3 Requirements Engineering & Management Support (Requirement) EX.1.4.1 Requirements Engineering Architecture (Requirement) MC.1.1.3.1 Requirements & Architecture Support (Requirement) EX.1.4 Customer Requirements Support (Requirement)
PLMLog.2.16 Systems Engineering Team (Logical Architecture)	MC.1.1.2 Production Configuration Management (Requirement) MC.1.1.3 Production Systems Engineering Support (Requirement) INT.1.1.4 Communicate Conceptual Systems Engineering Architecture (Requirement) EX.1.3.1 Implement System Architecture (Requirement) EX.1.3 System Architecture (Requirement) MC.1.1.3.1 Requirements & Architecture Support (Requirement) NWP.1.2 Systems Engineering Support (Requirement) NWP.1.4 Configuration Management Support (Requirement) INT.1.1.1.1 Conceptual Configuration Management Architecture Plan (Requirement) NWP.1.3 Requirements Engineering & Management Support (Requirement) INT.1.1.3 Communicate Conceptual Requirements Engineering Plan (Requirement) EX.1.2 Product Definition Item (Requirement) EX.1.2.1 Implement Product Definition Architecture (Requirement) INT.1.1.2.1 Product Definition Architecture Plan

Allocated Capabilities/Requirements	Traced From Higher-Level Blocks
	(Requirement) INT.1.1.4.1 Systems Architecture Plan (Requirement) INT.1.1.3.1.1 Accept/Reject Requirements Engineering Architecture (Requirement) EX.1.1.1 Implement Configuration Management (Requirement) EX.1.4 Customer Requirements Support (Requirement) EX.1.1 Configuration Management Items (Requirement) INT.1.1.3.1 Requirements Engineering Architecture Plan (Requirement)
PLMLog.3.1 Computer Aided Design & Drafting Technologist (Logical Architecture)	INT.1.1.1 Communicate Conceptual Configuration Management Plan (Requirement) QC.1.2.2.1 Perform Models & Drawing Audits (Requirement) INT.1.1.1.1.1 Accept/Reject Configuration Management Architecture (Requirement) EX.1.5.1 Manage Access Request (Requirement) MC.1.2 Production Software/Hardware/IT Support (Requirement) CL.1.1.1 Legacy Part Definition Maintenance & Closing Support (Requirement) NWP.1.4 Configuration Management Support (Requirement) EX.1.5 Accept Access Request (Requirement) INT.1.1.1.1 Conceptual Configuration Management Architecture Plan (Requirement) MC.1.1.1.1 Production Product Realization (Requirement) QC.1.2.1.1 Perform Configuration Management Audits (Requirement) EX.1.1.1 Implement Configuration Management (Requirement) NWP.1.6 Computer Aided Drafting & Design Support (Requirement) MC.1.1.2 Production Configuration Management (Requirement)
PLMLog.3.2 Engineering Document Control Technologist (Logical Architecture)	NWP.1.8 Product Structure Support (Requirement) QC.1.2.2 Audit Product Production Definition Architecture (Requirement) CL.1.1 Product Definition & Configuration Management (Requirement) CL.1.1.1 Legacy Part Definition Maintenance & Closing Support (Requirement) MC.1.1.1.1 Production Product Realization (Requirement) CL.1.1.1.2 Retirement & Record Archiving

Allocated Capabilities/Requirements	Traced From Higher-Level Blocks
	(Requirement) CL.1.1.1.1 Material List/Procurement Index Maintenance (Requirement) EX.1.2 Product Definition Item (Requirement) EX.1.2.1 Implement Product Definition Architecture (Requirement) INT.1.1.2 Communicate Conceptual Product Definition Architecture (Requirement) QC.1.1.1 First Production Product Definition Maintenance Quality Control (Requirement) INT.1.1.2.1 Product Definition Architecture Plan (Requirement) MC.1.1.1 Production Product Structure Support (Requirement)
PLMLog.3.3 Engineering Program/Project Lead (Logical Architecture)	INT.1.1.1 Communicate Conceptual Configuration Management Plan (Requirement) QC.1.2.1 Audit Configuration Management Architecture (Requirement) QC.1.2.2.1 Perform Models & Drawing Audits (Requirement) QC.1.2.2 Audit Product Production Definition Architecture (Requirement) NWP.1.7.1 Computer Network Capabilities (Requirement) MC.1.2 Production Software/Hardware/IT Support (Requirement) EX.1.8 Product Structure Evolution Process (Requirement) QC.1.2 First Production Customer Quality Control (Requirement) NWP.1.1 Support Databases (Requirement) EX.1.9 Product Lifecycle Management Change Process (Requirement) INT.1.1.1.1 Conceptual Configuration Management Architecture Plan (Requirement) EX.1.8.1 Change Advisory Board (Requirement) MC.1.2.1 Change Advisory Boards (Requirement) NWP.1.5 Accessing Nuclear Weapon Need (Requirement) QC.1.1.1.1 Audit Ticketing Process (Requirement) QC.1.2.1.1 Perform Configuration Management Audits (Requirement) EX.1.7 Product Lifecycle Management IT Support (Requirement) NWP.1.6 Computer Aided Drafting & Design Support (Requirement) NWP.1.7 Establishing Customer Computer Network

Allocated Capabilities/Requirements	Traced From Higher-Level Blocks
	Support (Requirement)
PLMLog.3.4 Engineering Systems Integration/Implementation Pro (Logical Architecture)	QC.1.2.1 Audit Configuration Management Architecture (Requirement) QC.1.2.2.1 Perform Models & Drawing Audits (Requirement) QC.1.2.2 Audit Product Production Definition Architecture (Requirement) NWP.1.7.1 Computer Network Capabilities (Requirement) MC.1.2 Production Software/Hardware/IT Support (Requirement) QC.1.2 First Production Customer Quality Control (Requirement) NWP.1.4 Configuration Management Support (Requirement) NWP.1.1 Support Databases (Requirement) EX.1.9 Product Lifecycle Management Change Process (Requirement) EX.1.8.1 Change Advisory Board (Requirement) MC.1.2.1 Change Advisory Boards (Requirement) QC.1.1.1.1 Audit Ticketing Process (Requirement) QC.1.1.1 First Production Product Definition Maintenance Quality Control (Requirement) QC.1.2.1.1 Perform Configuration Management Audits (Requirement) EX.1.7 Product Lifecycle Management IT Support (Requirement) NWP.1.6 Computer Aided Drafting & Design Support (Requirement) NWP.1.7 Establishing Customer Computer Network Support (Requirement)
PLMLog.3.5 Information Systems Security Technologist (Logical Architecture)	EX.1.9 Product Lifecycle Management Change Process (Requirement) EX.1.8.1 Change Advisory Board (Requirement) NWP.1.1 Support Databases (Requirement) NWP.1.7 Establishing Customer Computer Network Support (Requirement) EX.1.7 Product Lifecycle Management IT Support (Requirement) MC.1.2.1 Change Advisory Boards (Requirement) QC.1 Product Lifecycle Management Quality Control (Requirement) MC.1.2 Production Software/Hardware/IT Support (Requirement) NWP.1.7.1 Computer Network Capabilities (Requirement) EX.1.5 Accept Access Request (Requirement)

Allocated Capabilities/Requirements	Traced From Higher-Level Blocks
PLMLog.3.6 Manager, R&D Science & Engineering (Logical Architecture)	CL.1 Product Lifecycle Management Architecture Closing Support (Requirement) NWP.1 Product Lifecycle Management Architecture Authorization (Requirement) MC.1 Product Lifecycle Management Architecture Monitoring & Controlling (Requirement) EX.1 Product Lifecycle Management Execution(Requirement) QC.1 Product Lifecycle Management Quality Control (Requirement) INT.1 Product Lifecycle Management Initiation (Requirement)
PLMLog.3.7 Systems Engineer (Logical Architecture)	CL.1 Product Lifecycle Management Architecture Closing Support (Requirement) INT.1.1.1 Communicate Conceptual Configuration Management Plan (Requirement) INT.1.1.4 Communicate Conceptual Systems Engineering Architecture (Requirement) EX.1.3.1 Implement System Architecture (Requirement) EX.1.3 System Architecture (Requirement) EX.1.4.1 Requirements Engineering Architecture (Requirement) MC.1.1.2 Production Configuration Management (Requirement) MC.1.1.3 Production Systems Engineering Support (Requirement) MC.1.1.3.1 Requirements & Architecture Support (Requirement) INT.1.2 Communicate Nuclear Weapon Conceptual Design (Requirement) NWP.1.2 Systems Engineering Support (Requirement) QC.1 Product Lifecycle Management Quality Control (Requirement) MC.1.1 Production Customer Support (Requirement) NWP.1.4 Configuration Management Support (Requirement) NWP.1.3 Requirements Engineering & Management Support (Requirement) INT.1.1.3 Communicate Conceptual Requirements Engineering Plan (Requirement) EX.1.2 Product Definition Item (Requirement) EX.1.2.1 Implement Product Definition Architecture (Requirement) INT.1.1.2.1 Product Definition Architecture Plan (Requirement) INT.1.1.4.1 Systems Architecture Plan (Requirement)

Allocated Capabilities/Requirements	Traced From Higher-Level Blocks
	EX.1.1.1 Implement Configuration Management (Requirement) EX.1.4 Customer Requirements Support (Requirement) INT.1.3.1.1 Deliver Revised Conceptual Architecture Documents (Requirement) EX.1.1 Configuration Management Items (Requirement) INT.1.1.3.1 Requirements Engineering Architecture Plan (Requirement)
PLMLog.3.8 Solutions Architect (Logical Architecture)	CL.1 Product Lifecycle Management Architecture Closing Support (Requirement) INT.1.1.1 Communicate Conceptual Configuration Management Plan (Requirement) INT.1.1.4 Communicate Conceptual Systems Engineering Architecture (Requirement) EX.1.3.1 Implement System Architecture (Requirement) EX.1.3 System Architecture (Requirement) EX.1.4.1 Requirements Engineering Architecture (Requirement) MC.1.1.2 Production Configuration Management (Requirement) MC.1.1.3 Production Systems Engineering Support (Requirement) MC.1.1.3.1 Requirements & Architecture Support (Requirement) INT.1.2 Communicate Nuclear Weapon Conceptual Design (Requirement) NWP.1.2 Systems Engineering Support (Requirement) QC.1 Product Lifecycle Management Quality Control (Requirement) MC.1.1 Production Customer Support (Requirement) NWP.1.4 Configuration Management Support (Requirement) NWP.1.3 Requirements Engineering & Management Support (Requirement) INT.1.1.3 Communicate Conceptual Requirements Engineering Plan (Requirement) EX.1.2 Product Definition Item (Requirement) EX.1.2.1 Implement Product Definition Architecture (Requirement) INT.1.1.2.1 Product Definition Architecture Plan (Requirement) INT.1.1.4.1 Systems Architecture Plan (Requirement) EX.1.1.1 Implement Configuration Management (Requirement) EX.1.4 Customer Requirements Support (Requirement)

Allocated Capabilities/Requirements	Traced From Higher-Level Blocks
	INT.1.3.1.1 Deliver Revised Conceptual Architecture Documents (Requirement) EX.1.1 Configuration Management Items (Requirement) INT.1.1.3.1 Requirements Engineering Architecture Plan (Requirement)